MAGAZINE VOOR
ATARI - ST
GEBRUIKERS

NUMMER 7 SEPTEMBER / OKTOBER 1987 PRIJS F6.95 / Bfr 140

# START

Aladin
Welcome to Macintosh

Bliksembezoek
MEGA ST 4

Weg met de desktop accessoires!

Sprite probleem opgelost

COCKPIT



**START** een uitgave van
Divo computerservice
M. Gijzenburg 14
2907 HG Capelle a/d IJssel
Tel. 010 - 458 76 40

**REDAKTIE:**
Jan van Die
Hannie van Leeuwen
Ieke Looyé
Patrick Wessels

**VORMGEVING:**
Studio Wessels & Van Leeuwen
C.P. Tielestraat 3a
3023 TA Rotterdam
Tel. 010 - 476 98 98

**ADVERTENTIE-EXPLOITATIE:**
Hannie van Leeuwen
Tel. 010 - 476 98 98

**MEDEWERKERS:**
Frank Abbing
Mark Boon
Daan Josephus Jitta
Heino Kaal
Alfred Konijnenbelt
Robert van der Kroft
René van der Linden
Egon Lubbers
Karl Moens
Wilbert Plijnaar
Auke Sikma
Markus The
Jos Vermaseren
Peter Witvoet
Piet Zeeman

**ABONNEMENTEN:**
Jaarabonnement (6 nummers)
Maak f 39,95 over naar
postgiro 1962230
t.a.v. Divo computerservice/START
M. Gijzenburg 14
2907 HG Capelle a/d IJssel
Abonnementen kunnen elk nummer ingaan en worden jaarlijks automatisch verlengd.

Opzeggingen moeten twee maanden voor het verstrijken van het abonnementsjaar schriftelijk ingediend worden.

**ZETWERK:**
De Stad b.v. Delft
Tel. 015 - 12 46 79

**DRUK:**
Van den Berg Zwijndrecht
Tel. 078 - 10 09 11

**VERSPREIDING:**
Betapress Gilze
Tel. 01615 - 78 00




# VAN DE FEEST REDAKTIE

***HOERA! WE BESTAAN EEN JAAR!***



*En omdat feestjes vieren in je eentje niet leuk is, hebben alle START-abonnees bij dit nummer een kwaliteitsdiskette van BASF gevonden met daarop alle listings uit het blad. Dat is wel even een financieel rib uit ons lijf, maar zonder al die mensen die vol vertrouwen f 39,95 naar een beginnend blad durfden te storten, waren we er nu niet meer geweest.*
*Maar op je verjaardag doe je niet kinderachtig. Wie in september of oktober alsnog abonnee wordt, krijgt de diskette ook! De produktie van deze schijf is mede mogelijk gemaakt door een financiële bijdrage van de firma P.D.S. Wie nu denkt dat die schijf verder vol met reklame staat, heeft het mis. "Zie het als een verjaarskado en zet er op wat je wilt", kregen we te horen. U begrijpt dat we ons dat geen twee keer lieten zeggen.*
*Buiten de listings uit dit nummer bevat onze verjaardagsschijf dan ook nog wat extra verrassingen. Op het moment dat we dit schrijven, zijn we nog druk bezig om een aantal representatieve programma's te verzamelen. We vertrouwen erop dat onze selektie bij u in de smaak zal vallen.*
*Tot slot een woord van dank aan de P.T.T. Het is in Nederland eigenlijk (nog) niet toegestaan om diskettes bij tijdschriften te verpakken. Vanwege onze verjaardag kregen we een eenmalige ontheffing. START is daardoor het eerste Nederlandse computertijdschrift dat zijn abonnees zo kan verwennen. Van ons hoort u voorlopig geen ambtenarenmoppen!*

De Redaktie

# INHOUD

# DE MARKT

**Op het moment dat we dit schrijven, is het wat stil aan het softwarefront. De top van de verkopen verschuift steeds meer naar het einde van het jaar. Reden dat veel premières opgehouden worden. Dat geeft ons de tijd om een aantal produkten wat intensiever te bekijken dan u normaal in deze rubriek gewend bent.**

## VEEL KODE

Importeur Panatco zond ons twee exemplaren van FaSTer, een Amerikaans tijdschrift op diskette. Hoewel de redaktie in haar editorial van volume 2.2 wat zeurt dat FaSTer door de tijdschriften wordt doodgezwegen, schrikken wij niet voor een beetje konkurrentie.

Op volume 2.1 treffen we drie grote programma's: een monitor en de spelletjes Othello en Reboink.

Volume 2.2 bevat een programma om ASCII-bestanden geformatteerd af te drukken en het kaartspel Scribbage.

Die programma's zijn de kracht van een magazine op disk. START zou ze niet kunnen publiceren. Zoveel kode tikt geen mens in.

De zwakte van de schijven zijn naar onze smaak de (Amerikaanse) teksten. Een onbekend artikel leest niet echt prettig op zo'n monitor. Je mist het overzicht en kunt niet makkelijk een stuk even herlezen. Ook irriteert het scrollen, daar de onderste regel telkens even dubbel in beeld is. Aan de andere kant kost de hele schijf afdrukken een flinke berg papier.

Cursussen op de diskettes gaan onder andere over C, Pascal, animatie en de floppy. Veel ruimte voor de bespreking van software. Met name de adventures komen flink aan bod. Amerika loopt wel een beetje achter: voor de Nederlandse markt is een vergelijking tussen GfA en Fast Basic niet echt aktueel meer.

Een beperking van het medium is de hoeveelheid beschikbare ruimte. FaSTer is geformatteerd op een manier die maar liefst 372.818 bytes op een enkelzijdig schijfje toelaat, maar met een paar lange programma's is ook dat snel op. Door het kleine aanbod is de kans groot dat FaSTer af en toe tegenvalt. De programma's zullen niet altijd de smaak van de koper volgen.

Een komplimentje voor de interface is op zijn plaats. Alle teksten en programma's zijn via hetzelfde (automatisch startende) menu te bereiken.

De prijs/prestatieverhouding van FaSTer is goed. B.T.W. en verzendkosten in aanmerking genomen is f 17,50 een koopje.

## MODULATOR VOOR BODEMPRIJS

Wie een ST op een TV wil aansluiten, is het beste uit met een scartkabel. Veel tv's bezitten hier helaas nog geen ingang voor. In dat geval brengt een modulator uitkomst. De STM-modellen zijn standaard met zo'n modulator uitgerust. Voor de andere ST's brengt de firma Brunelco een losse modulator uit.

Wat de prijs betreft: een weggevertje. Atari rekent al f 113,- voor een scartkabel. Voor f 149,- krijgt u heel wat meer elektronika.

De modulator beschikt bijvoorbeeld over een eigen voeding en biedt naast een standaard tv-connector ook aparte uitgangen voor beeld en geluid. De modulator is keurig afgewerkt in een stevige behuizing.

De firma raadt gebruik aan in lage resolutie. In de middelste resolutie werkt de modulator overigens wel, maar de bandbreedte van de TV maakt een echt goed beeld niet mogelijk.

Een dezer dagen bereikte een exemplaar de redaktie. We hebben er een tijdje mee gewerkt en dat viel ons beslist niet tegen. Wel liep er een licht storingsstreepje door het beeld en was er wat beweging in de kleuren te zien, maar elk vertaalproces kent nu eenmaal zijn prijs. Verder bleek het snoertje naar de ST aan de korte kant, maar waarschijnlijk is dit gedaan om verder signaalverlies te voorkomen.

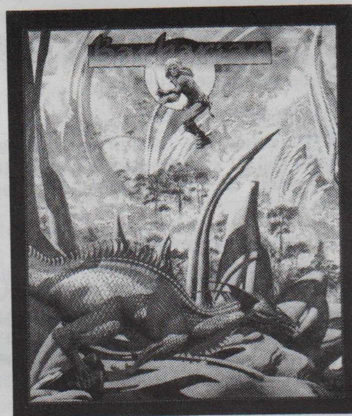
De redaktie denkt er hard over om een aantal van deze modulatoren aan te schaffen, zodat auteurs die bereid zijn hun 'monochrome' programma voor kleur aan te passen er tijdelijk één kunnen lenen.

Voor langdurig gebruik lijkt de modulator ons minder geschikt. Wie af en toe eens een spelletje wil spelen, gooit zijn geld echter beslist niet over de balk.

## BARBAARS SPEL

BARBARIAN, van de Engelse firma Psygnosis, is een kruising tussen een grafisch adventure en een aktiespel. Bij ouderwetse adventures moest je kommando's intikken voor de hoofdfiguur een vin verroerde. Nou ja, vin... Hegor, de held van het spel, staat eerder bol van spieren dan van vinnen. Je bestuurt hem door met de muis op ikonen aan de onderkant van het beeld te klikken.

Hegor bevindt zich in een vulkaan. Hij moet een kwaadaardig kristal opsporen, vernietigen en dan de kuierlatten nemen voor de vulkaan uitbarst.



Eerste indrukken: mooie plaatjes en heel speelbaar. Wel jammer dat je steeds weer naar het begin wordt gestuurd als een tegenstander Hegor om zeep heeft gebracht. Een klein stukje terug is ook straf hoor!

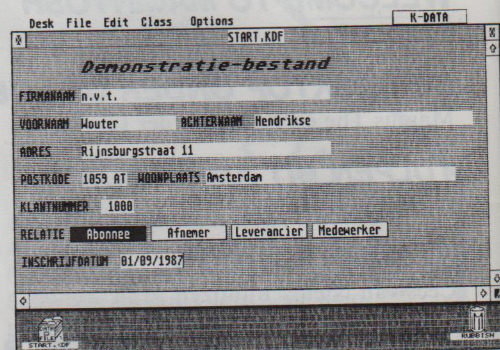
De software van Barbarian is nog niet helemaal ontluisd. Een week na het recensie-exemplaar arriveerde een eerste update op de redaktie. Na de tweede keer spelen, zat de computer muurvast.

We houden de brievenbus goed in de gaten: als je tijdens het laden met de muis speelt, krijg je bommen. En daar is zelfs Hegor de Barbaar niet tegen bestand.

## SEKSE KNOPPEN

KUMA SOFTWARE blijft haar uitgekiende reeks produkten uitbreiden. Iemand zou de firma eens moeten vertellen dat steeds hetzelfde beeldmerk op de verpakking geen verkoopbevorderend effekt heeft. Wie naar de inhoud van de bekende blauwe klappers doorstoot, treft echter meestal een heel behoorlijk produkt.

Dat is ook weer onze indruk van K-DATA. Een GEM-gestuurde database die uiteraard weer volledig kompatibel is met de andere K-produkten. Een positieve opmerking over het handboek is allereerst op zijn plaats. Meestal stelt dat bij Kuma uiterlijk niet al te veel voor. Het bedrijf wringt zich in allerlei bochten om toch maar alles in de standaard klapper te krijgen. Voor K-DATA is de klapper echter een doosje geworden: dat biedt plaats aan een stevig spiraalgebonden handboek.




K-DATA bestaat uit twee programma's. Het eerste wordt gebruikt om de layout van een database te maken; het tweede manipuleert de gegevens.

Twee tot vier bestanden kunnen tegelijk open zijn. De lengte van de bestanden is afhankelijk van de gebruikte hardware. Een veld kan maximaal 1600 karakters bevatten. Een record maar liefst 10.000. Een veld kan een default-waarde krijgen. Bijvoorbeeld handig als een firma bijna al zijn klanten in dezelfde plaats heeft wonen. Ook velden met 'knoppen' zijn mogelijk. In het veld 'sekse' kan de gebruiker dan bijvoorbeeld op 'man' of 'vrouw' klikken.

K-DATA lijkt een geschikt produkt voor mensen die een eenvoudig te bedienen databasemanager met een redelijk aantal mogelijkheden zoeken. Er zijn krachtiger programma's op de markt, maar die laten zich niet zo makkelijk bedienen.

## LYRISCH

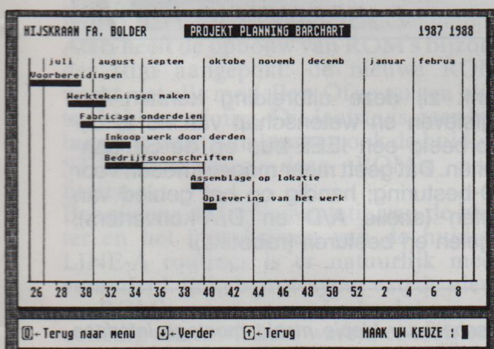
PDS heeft een uniek tekentablet voor de Atari ST. In het vorige nummer kondigden we deze digitizer al aan, maar nu hebben we er zelf ervaring mee opgedaan. Patrick Wessels, elk nummer verantwoordelijk voor de layout van START: "Zo'n muis maakt je hand, die immers een pennetje gewend is, ineens heel log. Het tekentablet laat je weer schetsen zoals je dat heel je leven gedaan hebt. Het enige verschil is nu nog dat de tekening niet onder de punt van je pen, maar op het beeldscherm ontstaat. Maar daar ben je zo aan gewend."

Ook onze vaste tekenaar Wilbert Plijnaar (zie onze voorplaat) was lyrisch over het tablet. Eindelijk begint ook hij wat in een computer te zien. We hebben het idee dat het tablet op die manier heel wat kreatieve mensen over de streep zal trekken.

Het programma werkt met bijna alle bestaande tekenpakketten samen. Alleen STAD gaf problemen, maar volgens de distributeur was ook daar al weer een software-mouw aan gepast.

## PLANNEN VIA ST

Peecee Software bracht ons de allereerste Nederlandse financiële software: de pakketten ST FINANCIAL JUNIOR, PERSONAL en VOORRAAD. De pakketten werden gedistribueerd door PDS.
TARGET is het eerste programma dat PeeCee onder eigen naam uitbrengt. Het pakket is bedoeld als hulpmiddel bij de planning en registratie van projekten. Zowel de beschikbaarheid van produktiemiddelen als man- en machine-uren, materialen en direkte uitgaven als de begroting en de werkelijke bestedingen kunnen met TARGET vastgelegd worden. Daar de aktiviteiten die in het kader van een projekt verricht moeten worden veelal afhankelijk van elkaar zijn, moeten ook zogenaamde netwerkrelaties worden opgegeven. Als er een verandering in een deel van het produktieproces optreedt, verzorgt TARGET aan de hand van deze relaties nieuwe calculaties en overzichten.
TARGET kan ook komplete projekten dupliceren. Wie vaak gelijksoortige werkzaamheden verricht, zal het waarderen dat alle basisgegevens niet telkens opnieuw ingevoerd hoeven te worden.
Het programma wordt geleverd met een keurige handleiding en wat voorbeeldbestanden om mee te experimenteren. Wat we missen in de handleiding is een voorbeeld waarbij we stap voor stap bij de hand worden genomen.
*Inlichtingen: PeeCee Software, Postbus 14, 9765 ZG Eelde.*




## BOEKBESPREKING:
### GfA BASIC

*Frank Ostrowski.*

Als je in GfA-BASIC programmeert, hou je natuurlijk een oogje open voor de informatie die hierover verschijnt. Een boek van Frank Ostrowski - de auteur van GfA-Basic - is dan een niet te missen informatiebron. Voor 99 gulden krijg je het boek van 287 pagina's en een diskette met de voorbeeldprogramma's uit het boek in je bezit. Alles (voorlopig nog?) in het Duits.
Het boek bevat de volgende onderdelen: programma's optimaliseren, grafische mogelijkheden, tips, GEMDOS, BIOS en XBIOS, AES, RSC en windows.
Het gedeelte over windows bevat ruim voldoende informatie en voorbeelden om een GfA-programma te schrijven dat op een korrekte manier van vensters gebruik maakt.
Een beginner zal met een aantal besproken onderdelen best moeite hebben. Vaak wordt er meer aangestipt dan echt uitgelegd. Met uitproberen kom je echter een heel eind.
Veel informatie die in dit boek voorkomt, is ook uit het grote GEM-boek en Atari ST Intern te halen. Het grote voordeel is dat alle informatie over de aanroepen hier bij elkaar geplaatst is. En voorbeelden bij het gebruik van de aanroepen werken ook verhelderend.
Kortom, een boek dat bij een serieuze GfA-programmeur op de plank hoort te staan...

*Peter Witvoet*

## GFA-STARTER

Zoals u wellicht weet, is het mogelijk om een programma automatisch te laten beginnen op het moment dat u de computer aanzet. Daar wordt dan ook hevig gebruik van gemaakt. Denk maar eens aan de ramdisk- en klokprogramma's die zich bij het opstarten installeren. Al deze programma's kunnen echter geen gebruik maken van de GEM-mogelijkheden, om de eenvoudige reden dat GEM bij het opstarten nog niet aanwezig is.
Om GEM-programma's toch automatisch te kunnen laten beginnen, is een speciaal programma nodig. GfA-STARTER is zo'n programma. Via een meegeleverde simpele editor kan de gebruiker opgeven hoeveel en welke programma's automatisch opgestart moeten worden. Hij moet daarvoor wel een AUTO-folder op de boot-schijf zetten met het programma STARTER.PRG erin! Het editor-programma maakt een tekstfile aan met de juiste gegevens.
Er kunnen maximaal tien programma's NA ELKAAR 'gerund' worden. Dit gaat als volgt: wanneer het eerste programma klaar is, wordt automatisch het tweede programma gestart. Dat gaat zo door tot alle programma's aan de beurt zijn geweest.
Tijdens het automatisch starten wordt een scherm getoond waarop de auteurs en het copyright vermeld staan. Jammer dat hier niet iets beters van is gemaakt. Het gebrek aan vormgeving werkt enigszins storend. Om bekendheid te krijgen bestaan er andere methodes.
Op GfA-STARTER berust copyright. Het is dus niet toegestaan om dit programma op een schijf die u weggeeft of verkoopt te zetten. U zult dan een regeling met de auteurs of de uitgever moeten treffen.
De prijs van 79 gulden is vrij hoog te noemen. Een gunstiger prijsstelling zou beter in verhouding staan met de toepassing van GfA-STARTER. Per slot van rekening wordt het programma alleen gebruikt voor een - klein - stukje gemak.

*Peter Witvoet*

## DE KINDEREN VAN DE WIND

*Graphic adventure gebaseerd op de gelijknamige strip door Francois Bourgeon.*

Een doorbraak? Nou, kom, nee, laten we niet overdrijven. Het is tenslotte maar een spelletje. Maar het is wel het allereerste spelletje gebaseerd op een bestaande realistische film of strip, waarin het origineel niet op gruwelijke wijze is mishandeld. De beelden op het scherm lijken echt op de tekeningen in de strip, en dat is nieuw.
We hebben te maken met een graphic adventure: plaatjes met praatjes. Je hoeft als speler overigens niet zelf teksten in te typen, zoals bij de ouderwetse adventures. Je zet gewoon de cursor op de persoon van je keuze - afhankelijk van het getoonde tafereel - en vervolgens verschijnt zijn of haar tekst in een box onderin het beeld. Als dit een voorstel aan de andere aanwezigen is ("Zullen we die schurk de hersens inslaan?") dan kun jij weer kiezen uit een aantal reakties van de rest van het gezelschap. (Kies tussen: "Nou, ik weet het niet, volgens mij krijgen wij daar last mee", en "Geef hier die hamer!"). De speler heeft een grillig wegennet van foute, goede en minder goede keuzes voor zich om de hoofdpersonen naar een happy end te leiden.




Gepuzzel derhalve, vergezeld van wondermooie beelden. Alleen jammer dat de importeur een chimpansee heeft gehuurd om de Franse tekst van het programma in het Nederlands te vertalen. Maar je krijgt een mooi stripboek bij aankoop van het spel, zodat je kunt lezen hoe de teksten ongeveer hadden moeten zijn. Laten wij niet klagen, er is genoeg echte troep op de markt. "De kinderen van de wind" is een boeiend en gewetensvol gemaakt spel. In Frankrijk schijnen ze trouwens al met deel twee bezig te zijn; met zes stripalbums in de winkel ziet het er dus naar uit dat wij nog het een en ander kunnen verwachten.

*Piet Zeeman*

## • KORTE BERICHTEN

De heer C. Lai is één van de oprichters van een nieuwe ST computerclub. Het is de bedoeling van de club om regelmatig een 'blad op schijf' uit te geven. De kontributie bedraagt f 10,-.
*Inlichtingen: ST CLUB INTERNATIONAL, Postbus 13672, 2501 ER 's-Gravenhage.*

Ariolasoft houdt het voor gezien. De importeur die onder andere produkten van Activision en Batteries Included (DEGAS) naar Nederland haalde, ziet verkoop van software kennelijk niet langer als een winstgevende zaak. Een veeg teken. Maar aan de andere kant: als dat zo door gaat, moeten de illegale kopieerders op den duur de software die ze zoeken zelf programmeren...

ABC Software is de importeur geworden van twee netwerksystemen voor de ST: Punchnet en Meganet.
Punchnet is vooral geschikt voor kleinere bedrijven en het onderwijs. Meganet is een flink stuk duurder, maar beschikt dan ook over een eigen processor waarmee een hoge verwerkingssnelheid valt te behalen.
*Inlichtingen: ABC Software, Dorpsstraat 2, 5314 AE Bruchem.*

Atari Benelux probeert de ontwikkeling van edukatieve software zoveel mogelijk te stimuleren. Naast de zojuist afgesloten programmeerwedstrijd is er nu onder de naam 'rekencircus-ST' een schijfje naar alle dealers gezonden. Deze mogen het gratis aan de klanten doorgeven. Het is een aardig vormgegeven programma waarin één of twee clowntjes in een touw klimmen om een zwevend parapluutje te pakken te krijgen. Het klimmen wordt 'gevoed' door korrekt opgeloste rekensommen.

# Bliksembezoek MEGA ST4

**Net nadat alle artikelen voor dit nummer ingeleverd waren, kregen we een telefoontje van Atari: er stond een Mega ST4 voor ons klaar. We mochten hem een week van alle kanten bekijken.**
**In zo'n geval zit er maar één ding op: in de auto springen, het apparaat ophalen en de zetter en layouter laten weten dat er nog meer werk op komst is. Bijkomend voordeel: je leert een hoop nieuwe scheldwoorden.**



De Mega ST4 zou zonder blitterchip komen. Dat stond al in ons vorige nummer, dus aan dat idee waren we intussen gewend geraakt. Natuurlijk is er al heel veel over de Mega geschreven. Het meeste daarvan was gebaseerd op persdokumentatie en even aanraken op beurzen. Nu mochten we alle programma's die we hadden op de machine loslaten en op ons gemak de printplaat bestuderen. We vragen nogmaals uw aandacht voor de Mega ST, maar nu gebaseerd op eigen ervaringen!

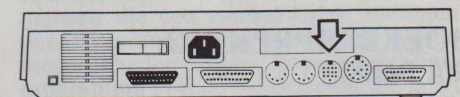
In positieve zin viel allereerst de handleiding op. Die is vollediger en beter gerubriceerd. Men heeft duidelijk de ervaringen met de problemen van beginnende gebruikers verwerkt. Dan gauw naar de computer. Even een tekstje tikken maakte duidelijk dat er iets is veranderd aan het toetsenbord. Het oogt hetzelfde, maar heeft een prettiger aanslag. De toetsen geven wat meer 'tegendruk'. Per abuis twee toetsen tegelijk aanslaan kost daardoor meer moeite.
De Mega maakt een einde aan de snoeren en transformatoren wirwar. Er is één stekker voor de hele computer inklusief diskdrive en de voeding is ingebouwd.
Dan natuurlijk de vier megabyte geheugen: ruim voldoende voor zelfs het meest veeleisende programma. En de mogelijkheid om een ramdisk van bijvoorbeeld een megabyte te installeren, is natuurlijk ook likkebaarden.
Een klok met batterij, node gemist bij de oudere ST's, is nu van de partij. Hij wordt gevoed door twee AA-batterijtjes die op een bijzonder bereikbare plaats in de systeemkast passen: bovenop. Bijna onzichtbaar weggewerkt in de linker koelopeningen.

## BINNENSTE BUITEN

De Atari ingenieurs hebben een prachtige printplaat afgeleverd. De onderdelen hebben alle ruimte. Dat is prettig voor het verdere moment dat er service gegeven moet worden. Natuurlijk zijn de meeste onderdelen bekende kost. Alleen de vier groepen van acht Megabit chips zijn nieuw. En er is al een voetje voor de blitterchip. Het kan dus niet meer misgaan: hij wordt er straks echt bijgeprikt. De chip krijgt dezelfde vierkante vorm als de GLUE en MMU-chips.
Oog voor detail kan de ontwerpers niet ontzegd worden. Zo vinden we bijvoorbeeld klemmetjes die de GLUE en de MMU naar beneden drukken. Opvallend is ook de zware afschermplaat. Die zal heel wat interferentie voorkomen.
Er is ook een nieuwe aansluitmogelijkheid: de 68000 is rechtstreeks naar buiten gevoerd. De 64-polige konnektor zit helaas niet aan de achterkant van de systeemkast, maar midden op de printplaat. Wel bevat de achterzijde een uitschuifbaar klepje, zodat gemakkelijk een interface aangebracht kan worden. Dat is wel nodig ook, want zo te zien is de konnektor niet gebufferd.

Dank zij deze uitbreiding kunnen bedrijfsleven en wetenschap van nu af bijvoorbeeld een IEEE-bus op de ST aansluiten. Dat geeft meer mogelijkheden voor I/O-besturing; handig op het gebied van meten (snelle A/D en D/A-konverters), regelen en besturen (robotika).

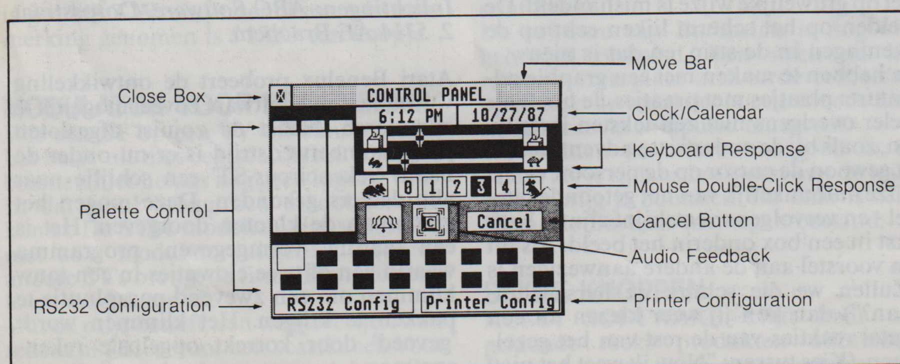
*Uitschuifbaar klepje maakt montage interface mogelijk.*



De printplaat bevat ook zes nieuwe ROM's, al is daar zonder blitter weinig van te merken. Alleen de toegevoegde alertboxen 'Save desktop ? - OK/Cancel' en 'Print screen display ? - OK/Cancel' wijzen op een verandering. De gebruiker kan die mogelijkheden dus bijna niet meer per ongeluk aktiveren.
Overigens is er wel een model MET blitter in Nederland. Dat staat (uiteraard) bij Atari zelf. Wilfried Kilwinger, onze informatiebron binnen het bedrijf, doet hierna verslag van zijn bevindingen met de ROM's en de blitter, dus maaien we hier het gras verder niet voor zijn voeten weg.

## WAT JE BELOOFT...

Een paar tegenvallers moeten ons ook van het hart. Wat de hardware aangaat, betreft dat eigenlijk alleen de ventilator. Die is maar heel klein, waardoor hij heel wat toeren moet maken om lucht te verplaatsen. En dat hoor je!
In de handleiding van de Mega ST staat een afbeelding van een nieuw kontrole



paneel, waarin 'RS232 Config' en 'Printer Config' geïntegreerd zijn. Een goed idee! Hoe minder accessoires standaard geladen moeten worden, hoe flexibeler de gebruiker zijn desktop kan inrichten. Helaas ontbrak dit handige accessoire nog in het pakket.
Tenslotte was de aanvullende handleiding van de lang verwachte 'update' van ST BASIC bijgevoegd. Het ingesloten schijfje bevatte echter nog steeds de oude versie. Overigens zal het bedrijfsleven volgens ons niet op ST BASIC zitten te wachten. Helemaal niet bijleveren lijkt ons een betere zet.
Het kan natuurlijk zijn dat Atari de primeur van deze bijprodukten aan de koper in plaats van de journalist wil geven, maar dat lijkt ons niet zo'n slimme politiek. Elke verslaggever die minder krijgt dan de handleiding vermeldt, gaat daarover schrijven.
Als de zakelijke markt de ST nu nog niet in de armen sluit, weten we het niet meer: sterk verbeterde vormgeving, enorm geheugen, onovertroffen beeldscherm, veel aansluitbussen en een scherpe prijs. Bovendien is er op elk gebied inmiddels software die even goed of zelfs beter is dan de konkurrende pakketten op de IBM-PC.
Tot slot gaat de tijd in het voordeel van de ST tellen. Het bedrijfsleven vindt het plezierig als een computerserie een lange adem heeft. Dat geeft het geruststellende gevoel dat service beschikbaar blijft. Atari heeft met haar Mega's te kennen gegegeven dat die periode weer met een groot aantal jaren verlengd mag worden.
●

# DE NIEUWE **ROM'S!!!**

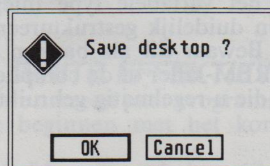
**Atari heeft zijn ST-reeks uitgebreid met de Mega-lijn: twee en vier Megabyte machines. Deze computers zijn op tal van punten verbeterd en uitgebreid en nieuw is onder andere de "blitterchip". Deze chip verzorgt grafische geheugenverplaatsingen en -bewerkingen: het zogenaamde bit-blitten. Voorheen gebeurde dit softwarematig. De daarvoor verantwoordelijke routines bevinden zich in ROM en zijn bij programmeurs bekend onder de naam LINE-A routines. Omdat de blitter de grafische bewerkingen hardwarematig aanpakt, was het nodig deze routines aan te passen. Het gevolg zijn nieuwe ROM's; speciaal voor de blitter. En als je toch nieuwe ROM's maakt, kun je maar beter gelijk alle op dat moment bekende fouten verbeteren. Wilfried Kilwinger probeerde een prototype van de nieuwe ROM's uit met maar liefst tweehonderd programma's**

## MINDER DISKWISSELS

Atari heeft de opbouw van ROM's bijzonder slim aangepakt: de nieuwe ROM werkt met alle modellen. Of ze nu een blitter hebben of niet. Dit laatste is zeer belangrijk, want dan kunnen ook de huidige ST-bezitters van deze ROM's profiteren.
Behoudens de implementatie van de blitter en het herschrijven van de huidige LINE-A routines is er natuurlijk meer veranderd. Ik heb prototypes van de nieuwe ROM's getest en, ondanks dat ze niet volledig uitontwikkeld waren, er toch een goed beeld van gekregen. De tests werden uitgevoerd met een Mega-4 inklusief blitter en een gewone 1040 zonder blitter.
Als we de machine met blitter opstarten, krijgen we in het desktop-menu onder "options" een extra regel met de tekst "blitter". Hiermee kunnen we desgewenst zelf de blitter aan- of uitschakelen. Op de 1040 zonder blitter kreeg ik deze regel niet te zien. Daar was dus uiterlijk niets aan de desktop veranderd.




Ik speelde wat met de funkties in het menu om te kijken of er iets gewijzigd was en inderdaad: bij "save desktop" kreeg ik een alertbox met OK en Cancel. Dat was nieuw en eigenlijk best wel handig. Ik vind het altijd een hoop werk om de desktop inklusief de te openen vensters op een prettige manier op het scherm te krijgen. Als ik later dan per ongeluk op "save desktop" druk, kan het feest weer opnieuw beginnen.
Naast deze verandering viel er in de menu's niets te ontdekken. Wel kwam ik er achter dat bij het kopiëren van A naar een virtual B-drive, er voortaan minder diskwissels nodig zijn. De check op de destination disk wordt in dat geval weggelaten. Dit betekent slechts één diskwissel per file in plaats van twee. Mensen die met één diskdrive werken, zullen dit waarschijnlijk met gejuich begroeten.




## SNELLER DISK I/O

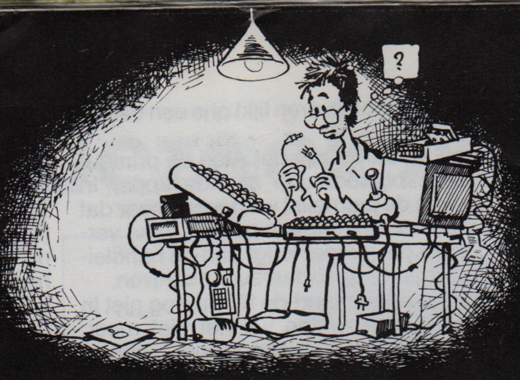
Omdat er vanwege de blitter toch nieuwe ROM's nodig waren, heeft ATARI de gelegenheid aangegrepen om een aantal fouten in het huidige operating system te verbeteren. Zo vinden we in de nieuwe ROM's oplossingen voor het RS-232 probleem en de welhaast "beroemde" 40-folder limiet. Ook minder bekende problemen in het GEMDOS en een stukje AES zijn opgelost. Ten tijde van deze test-ROM's waren nog niet alle bekende problemen ondervangen, maar een paar zeer vervelende fouten in ieder geval wel. Ook claimt ATARI (USA) dat in de uiteindelijke versie de file I/O een 20 tot 25% sneller zal worden. Er is wel een verklaring voor deze verbetering te vinden: momenteel schijnt de floppy-controller aan het eind van elke track een time-out te krijgen, waardoor de disk een extra wenteling moet maken om weer aan het begin van de nieuwe track te komen. Als dit opgelost is, kom je inderdaad aan een snelheidsverbetering van rond de 25%.

## COMPABILITEIT

De vraag die men bij deze nieuwe ROM's natuurlijk het meest stelt, is in hoeverre ze compatibel zijn met de oude ROM's. Om dit te weten te komen, testte ik alle tweehonderd programma's die ik bezit op de beide systemen uit. Van de tweehonderd programma's waren er slechts tien die problemen gaven. Acht van deze tien programma's waren spelletjes en dergelijke software munt in het algemeen niet uit in net programmeerwerk. De probleemgevallen liepen niet op de MEGA-4. Dit werd veroorzaakt doordat de programmeurs van de betrokken software rechtstreeks in het videogeheugen schreven en schijnbaar niet voorzien hadden dat dit geheugen in een twee of vier Megabyte ST op een ander adres zit.
Dit zelfde probleem vertoonde het programma Music Studio destijds toen de 1040 ST uitkwam. In eerste instantie werd deze kwaal opgelost met het programma MAKE520.PRG en hier kon ik het oplossen met MAKE1040.PRG.
De twee niet-spelletjes die met de nieuwe ROM's problemen hadden, waren Tempus (editor) en GfA Basic. Tempus werkte naar behoren, maar als je stopte kreeg je bommen en liep de zaak vast. Dit bleek veroorzaakt te worden door een fout in de exit-routines van Tempus. Deze fout komt er in het huidige operating system niet altijd uit en er is dan ook geen verklaring te vinden voor het feit dat het met de nieuwe ROM's wel problemen geeft.
Bij GfA Basic ligt de zaak weer anders: GfA maakt gebruik van een niet door ATARI gegarandeerd adres om zijn CTRL+SHIFT+ALT break-instruktie te detekteren. Dit adres ligt in de nieuwe ROM's ergens anders, dus dit vraagt om een update van GfA zelf. Het is te hopen dat ze een betere oplossing voor dit probleem vinden. Verder werkten zowel GfA Basic als de compiler naar behoren.

*Mijn algemene indruk van de nieuwe ROM's is dat ze kompakter opgebouwd zijn en dat een aantal zaken daardoor sneller werkt. Ik heb zelfs het idee dat het beeldscherm ook zonder blitter vlugger opgebouwd wordt. In dat geval heeft men bij het herschrijven van de routines de zaak wat efficiënter gemaakt. Het feit dat deze ROM's compatibel zijn met alle oude versies van TOS en GEM, versterkt het idee dat de ATARI-ingenieurs goed werk hebben afgeleverd.*

*Wilfried Kilwinger*



# GfA VRAAGBAAK

**Onze vaste rubriek met tips en truuks rond het populaire GfA BASIC. Heeft u iets ontdekt wat volgens u de moeite van het vermelden waard is? Stuur het op. Elke bijdrage die het blad haalt, wordt gehonoreerd met een START-diskette.**
**GfA VRAAGBAAK is ook de rubriek waarin u vragen kunt stellen. Onze medewerker Heino Kaal zal niet rusten voor hij de oplossing voor uw problemen gevonden heeft.**

## TIP 1: *PROGRAMMA'S MET KOMMENTAAR*

Is het u ook wel eens overkomen dat u een programma geschreven heeft en enige maanden later de rode draad kwijt bent? Wel, u bent niet de enige. Bij mij kwam dat hoofdzakelijk door mijn 'commodore 64' periode. Commodore Basic herkent slechts de eerste twee letters van de naam van een variabele of string. Duidelijke namen als Score$ en Schoon$ worden in zo'n geval door de computer als dezelfde variabele opgevat. Daardoor beperkte je je automatisch tot onzinnige namen als Aa$ en Ab$.
Met de komst van de ST-computers werd deze beperking gelukkig opgeheven: in GfA-basic kan de gebruiker een variabele- of stringlengte gebruiken die gelijk is aan een regel van de editor!
Zulke lange variabele- en stringnamen gebruiken is nogal overdreven, maar het geeft een indruk van de ruime mogelijkheden om duidelijke namen te kiezen. Dat komt de overzichtelijkheid van het programma ten goede, zodat er bij later gebruik van een listing nauwelijks tijd aan puzzelen verloren gaat.
Een andere manier om listings overzichtelijk te houden, is kommentaar toevoegen. Maar dat heeft zijn keerzijde. Het kommentaar neemt kostbaar geheugen in beslag en de programma's werken er iets langzamer door. Daarom vindt u in listing 1 een REM-killer, die werd geschreven door Hans en Iban van der Wal. Het programma leest een BESTAND.LST, haalt Rem, REM, ' en ! -kommentaar weg en maakt een BESTAND.COP. Uiteraard is het verstandig om voor toekomstig gesleutel een kopie van het programma MET kommentaar te bewaren. Maar u had toch altijd al een backup-schijfje, nietwaar?
Wie naast GfA-Basic de beschikking heeft over de GfA-compiler, hoeft het programma niet in te tikken. Die zorgt er namelijk zelf voor dat alle kommentaarregels verdwijnen. Bovendien worden langere variabele- en stringnamen kompakter opgeslagen en snoept het type integer (%) minder geheugen.
Met een eenvoudige test kunt u dit nagaan. Schrijf een klein programma en ga hierin als volgt te werk:
a) gebruik lange stringnamen (b.v. Aanpassing$="Test")
b) doe hetzelfde met variabelen van het type 'integer' (Aanpassing%=1024)
c) verwerk ook veel rem, ', of !-regels
d) compileer het programma onder de naam "TEST01.PRG".
e) klik nu vanuit de desktop het programma "TEST01.PRG" eenmaal aan. Kies vervolgens "Show Info" uit het desktop menu "File" en noteer het getal dat achter "Size in bytes" staat.
Laad nu de GFA-interpreter weer in en verkort de variabele- en stringnamen tot een of twee karakters (in ons voorbeeld 'Ap$="Test"' en 'Ap%=1024' enz). Denk erom dat u de waarde van de variabelen en strings NIET verandert.
Hierna verricht u opnieuw de handelingen zoals omschreven bij d) en e). Als naam van het programma kunt u nu "TEST02.PRG" gebruiken.
Zoals u kunt zien, zijn er geen bytes bijgekomen of afgegaan.
Haal nu alle kommentaarregels weg uit het "TEST01.BAS" programma en compileer opnieuw. U ziet: de lengte van het programma is weer niet aangetast.
Herhaal de test nogmaals, maar haal nu alle procenttekens weg. Hierdoor veranderen de integers in 'REAL' variabelen. Na compileren zal het programma GROTER zijn geworden.
De moraal van het verhaal: gebruik zoveel mogelijk het variabele type integer en schrijf een duidelijk gestruktureerd programma. Bewaar dit als backup en gebruik de REM-killer of de compiler voor de versie die u regelmatig gebruikt.

## TIP 2: *OVERZICHT NAAR PRINTER*

Paul van Duin uit Leiden schreef een brief waarin hij de volgende vragen afvuurde:
a) Hoe stuur ik in GfA-Basic een overzicht van alle files op diskette rechtstreeks naar de printer?
b) Hoe kan ik een RSC-file maken met behulp van de GfA-interpreter en -compiler?
De eerste vraag kan op verschillende manieren beantwoord worden:
1) - Dir"A:*.*" To "LST:" (print geen folders).
2) - Files "A:*.*" To "LST:" (print folders, file lengte, uren en datum).
3) - Het programma "DIRSORT" uit START 4 overtypen.
4) - Het programma "KATALOGER" bij de redaktie bestellen. (Excuses voor de sluikreklame.)

## TIP 3: *BASIC EN RSC*

De tweede vraag is een echte uitdaging. Eerst iets over de RSC-file: een RSC-file

```
' *********** LISTING 1 ***********
' ** R E M  -  K I L L E R **
' **   Hans en Iban van der Wal   **
' *********************************
'
@Scherm
Do
  @Main
  Clear !Schoonmaken (string)variabelen
Loop
'
Procedure Main
  @Selektie
  @Defmouse
  @Input_output
  Defmouse 0     !Terug na verandering =>
 in Input_output
Return
'
Procedure Scherm
  Hidem
  Deftext 1,0,0,13
  Deffill 0,2,8
  Box 5,5,634,395
  Box 7,7,632,393
  Text 64,80,"                  Dit is een u =>
tility-programma van"
  Text 64,96,"                       Hans en  =>
Iban van der Wal"
  Text 64,144,"Het programma werkt al =>
leen met files in ASCII-formaat.  "
  Text 64,160,"Voor GfA zijn dat de p =>
rogramma's welke met de Save,A of   "
  Text 64,176,"List-optie zijn wegges =>
chreven en dus een LST-extensie hebben."
  Text 64,192,"Er wordt gecontroleerd =>
 op:                                  "
  Text 64,208,"Rem, REM, ' en ! welke =>
 laatste twee specifiek voor GfA-basic"
  Text 64,224,"van toepassing zijn."
  Text 64,256,"Bij het programmeren d =>
ient u er op te letten dat het"
  Text 64,272,"uitroepteken altijd wo =>
rdt voorafgegaan door een spatie, dit "
  Text 64,288,"voor het onderscheid m =>
et de zgn. Boolean-variabele(n)."
  Repeat
  Until Mousek Or Len(Inkey$)
  Cls
  While Mousek
  Wend
  Showm
Return
'
Procedure Selektie
  Deffill 1,2,2
  Repeat
    Graphmode 1
    Prbox 157,33,482,61
    Graphmode 2
    Text 200,50,"Kies het te bewerken =>
 bestand..."
    Fileselect "A:\*.LST","",F1$
    Cls
  Until F1$="" Or Exist(F1$)
  If F1$=""
    Edit
  Endif
  F3$=F1$
  Hulp_pos%=0
  Pos%=0
  Do
    Hulp_pos%=Pos%
    Pos%=Instr(Pos%+1,F3$,"\")
    Exit If Pos%=0
  Loop
  F3$=Mid$(F3$,Hulp_pos%+1)
  Mid$(F3$,Len(F3$)-2)="COP"
  Prbox 157,33,482,61
  Text 165,50,"...dat moet worden weg =>
geschreven als..."
  Fileselect "A:\*.*",F3$,F2$
  Cls
  If F2$=""
    Edit
  Endif
Return
'
Procedure Input_output
  Open "I",#1,F1$
  Open "O",#2,F2$
  @Box(73)
  Deftext 1,0,0,13
  Graphmode 1
  Deffill 1,2,1
  Text 140,190,"Ogenblikje..."
  Box 138,204,501,220
  Box 120,150,520,250
  Box 122,152,518,248
  Box 123,153,517,247
  L=361/Lof(#1)            !Lengte van h =>
 orizontale balk/LOF
  M=140                    !Linkerkant v =>
 an de balk
  @No_line(False)
  While Not Eof(#1)
    Line Input #1,A$
```

```
    For T=M To M+L*(Len(A$)+2)     ! '+ =>
2' vanwege CR en LF
      Pbox T-1,205,T,219
      Line T+1,205,T+1,219
    Next T
    Add M,L*(Len(A$)+2)             ! '+ =>
2' vanwege CR en LF
    @Onderzoek_rem
    If A$<>""
      Print #2,A$
    Endif
    Inc K%
  Wend
  @No_line(True)
  Close #1
  Close #2
  Cls
  @Box(74)
  Alert 2,"Nog meer files te bewerken =>
?",2,"Ja|Nee",B%
  If B%=2
    Edit
  Endif
Return
'
Procedure Onderzoek_rem
  P%=Instr(A$,"Rem")
  If P%>0
    If P%=1 Or Left$(A$,P%-1)=Space$( =>
P%-1)   !Rem altijd 1e teken van regel
      A$=""
      Dec K%
    Endif
  Else
    P%=Instr(A$,"REM")
    If P%>0
      If P%=1 Or Left$(A$,P%-1)=Space =>
$(P%-1)  !Idem
        A$=""
        Dec K%
      Endif
    Else
      P%=Instr(A$,"'")
      If P%>0
        If P%=1 Or Left$(A$,P%-1)=Spa =>
ce$(P%-1) !Idem
          A$=""
          Dec K%
        Endif
      Else
        P%=0
        Do
          P%=Instr(P%+1,A$," !")
          If P%>0           !Lastig vanw =>
ege Boolean-Var.
            T%=1            !Concessie:  =>
Altijd spatie voor
            U%=0            !het uitroep =>
teken.
            Repeat
              If Mid$(A$,T%,1)=Chr$(34)
                Inc U%
              Endif
              Inc T%
            Until T%=P%
          Endif
          Exit If Even(U%) Or P%=0
        Loop
        If P%<>0
          Mid$(A$,P%+1)=Space$(Len(A$ =>
)-P%)
        Endif
      Endif
    Endif
  Endif
Return
'
Procedure Defmouse
  Let Mouse$=Mki$(8)+Mki$(8)+Mki$(1)
  Let Mouse$=Mouse$+Mki$(0)+Mki$(1)
  For T%=1 To 16
    Let Mouse$=Mouse$+Mki$(0)
  Next T%
  For T%=1 To 16
    Let Mouse$=Mouse$+Mki$(0)
  Next T%
  Defmouse Mouse$
Return
'
Procedure No_line(Flg%)
  Dpoke Intin,Flg%
  Dpoke Contrl+2,0
  Dpoke Contrl+6,1
  Vdisys 104
Return
'
Procedure Box(G%)
  Dpoke Gintin,320
  Dpoke Gintin+2,200
  Dpoke Gintin+4,0
  Dpoke Gintin+6,0
  Dpoke Gintin+8,120
  Dpoke Gintin+10,150
  Dpoke Gintin+12,400
  Dpoke Gintin+14,100
  Gemsys G%
Return
```

wordt over het algemeen gebruikt voor de informatie die een programma in dropdown-menu's en boxen op de gebruiker afvuurt. U kunt iets dergelijks in GfA-Basic simuleren op de manier die gedemonstreerd wordt in de listings 2 en 3.
In het voorbeeld ga ik ervan uit dat er een eenvoudig adresbestand opgebouwd moet worden. Listing 2 bevat de informatie: de teksten die op het scherm terecht moeten komen en de koördinaten van de gewenste positie.
Listing 3 leest deze gegevens in en drukt ze op het scherm af. Het voordeel van deze methode is dat er bij een eventuele verandering of vertaling alleen in de RSC-file gerommeld hoeft te worden.
De vraag van Paul roept meteen een andere op: kun je in GfA-Basic ook gebruik maken van een RSC-bestand zoals dat door een aantal zogenaamde resource construction sets wordt geproduceerd. In een volgend nummer van START hopen we hier aandacht aan te besteden.

*LISTING 2*

```
Restore Gegevens
Open "O",#1,"ADRES.RSC"
For I%=0 To 5
  Read A%,B%,A$
  Write #1,A%,B%,A$
Next I%
Close #1
Gegevens:
Data 2,10,Naam,2,11,Voornaam,2,12,Adr =>
es,2,13,Pcode
Data 2,14,Plaats,2,15,Telefoon
'
```

*LISTING 3*

```
Dim Test_1%(5),Test_2%(5),Test$(5)
Open "I",#1,"ADRES.RSC"
For I%=0 To 5
  Input #1,Test_1%(I%),Test_2%(I%)
  Input #1,Test$(I%)
  Print At(Test_1%(I%),Test_2%(I%));T =>
est$(I%)
Next I%
Close #1
```

## TIP 4: **WRITE EFFEKTIEVER**

Nu zijn we aan een paar programmeertips toe. We beginnen met het kommando "write".
Met "write" kunt u een belangrijke geheugenwinst behalen. Niet alleen op de diskette, maar ook in uw programma. Listing 4 geeft een voorbeeld om de theorie te verduidelijken.

Nadat u het programma heeft gerund, staat er het volgende op diskette:

```
"Kaal","Heino","p/a M. Gijzenburg 14","2
907 HG","Capelle a/d IJssel"<CR><LF>
```

Deze file is 70 bytes groot. Verandert u nu "Write #1" in "Print #1", dan wordt de file zo weggeschreven ('^' betekent een spatie):

```
Kaal^^^^^^^^^^^^Heino^^^^^^^^^^^p/a M. G
ijzenburg 14^^^^^^^^^^2907 HG^^^^^^^^^Ca
pelle a/d IJssel<CR><LF>
```

De file is nu 98 bytes groot geworden. Dat is 28 bytes meer dan de vorige file.
Als u "Write #1" heeft veranderd in "Print #1" krijgt u ook een foutmelding: nr. 26 ("End of File Reached": einde van de file; EOF). Dit komt omdat de informatie als een string is weggeschreven, in tegenstelling tot "Write #1" die er vijf strings van maakte.

## TIP 5: **DE KOMMA DE BAAS**

Het voorbeeldprogramma van listing 5 gebruikt de kommando's "Print #1", "Write #1", "Line Input #1" en "Input #1". De tekst in de tweede regel bevat een komma in een string en dat geeft problemen. Om het duidelijk te maken kunt u het programma het beste even intikken. Run het om te kontroleren of het werkt.
Zet nu voor de tweede en negende regel (Line Input) een rem-teken.
Regel 3 wordt nu: Write #1,"Dit is een tekst,", (de rest van de regel). Regel 10 wordt nu: Input #1,A$,B$,C$.
Run het programma opnieuw en u ziet dat er een puinhoop ontstaat. Dit wordt veroorzaakt door de komma in de string van regel twee. Die zorgt ervoor dat het programma nu twee strings ziet. Alleen het kommando "Line Input" is in staat om komma's te verwerken.

## TIP 6: **MEER GEKNIPPER**

In onze eerste GfA-vraagbaak stond een tip die betrekking had op het al dan niet laten knipperen van de cursor. Daar was bij nader inzien nog wel wat meer bij te vermelden. Hier is een aangevuld overzicht:

| | |
|---|---|
| Void Xbios(21,0) | cursor uit |
| Void Xbios(21,1) | cursor aan |
| Void Xbios(21,2) | cursor knipperen |
| Void Xbios(21,3) | cursor niet knipperen |
| Void Xbios(21,4,getal) | knippersnelheid instellen |
| A%=Xbios(21,5) | knippersnelheid ophalen |

*LISTING 4*

```
Open "O",#1,"adres.dat"
Write #1,"Kaal","Heino","p/a M. Gijze =>
nburg 14","2907 HG","Capelle a/d IJssel"
Close #1
Open "I",#1,"adres.dat"
Input #1,Naam$,Voor$,Adres$,Pcode$,Pl =>
aats$
Close #1
Print At(3,2);
Print Voor$'Naam$;At(3,3);Adres$;At(3 =>
,4);Pcode$'Plaats$
'
```

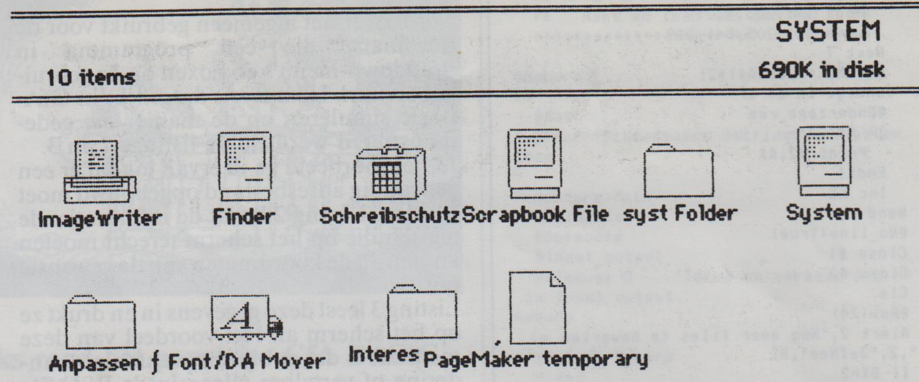
*LISTING 5*

```
Open "O",#1,"tekst"
Print #1,"Dit is een tekst,"
Write #1,"die wordt weggeschreven","n =>
aar diskette."
Close #1
' inladen van het bestand.
Open "I",#1,"tekst"
Line Input #1,A$
Input #1,B$,C$
Close #1
Print A$'B$'C$
```

---

*Ik hoop dat er tussen de informatie uit deze aflevering van de vraagbaak weer iets van uw gading was. Veel programmeerplezier en vergeet niet uw tips en truuks naar het onderstaande adres te zenden.*
*START REDAKTIE*
*GfA-rubriek*
*M. Gijzenburg 14*
*2907 HG Capelle a/d IJssel*

**Naast de Magic Sac van David Small is er nu ook een oerdegelijk Duits produkt op de markt: Aladin. Aladin is - net als de Magic Sac - een Macintosh emulator. Dit wil zeggen dat de ST-gebruiker uit de rijke bron van Macintosh software kan putten. Denk bijvoorbeeld aan 'Ready Set Go', 'Turbo Pascal', 'Excel', 'Comic Works' of 'Ragtime'.**




## DE UITDAGINGEN

De Magic Sac en Aladin zijn gebaseerd op hetzelfde principe. Zowel de Macintosh als de ST hebben de 68000 processor aan boord. Dat is dus meegenomen, want hierdoor kan de ST de Mac snel emuleren. Dit in tegenstelling tot de CP/M emulator - CP/M draait gewoonlijk met een andere processor: de Z80 van Zilog. Omdat er dan ook nog een andere processor nagebootst moet worden, werkt deze emulator niet al te snel.
Naast het feit dat de processor hetzelfde is, bleek tevens dat de systeemsoftware van de Macintosh grotendeels machine-onafhankelijk geschreven is. Grotendeels, maar niet helemaal. Het probleem is dat de onvermijdelijke hardware verschillen netjes moeten worden opgevangen. Hiervoor draagt een klein programmaatje zorg. De andere helft van de emulator is een printje met Macintosh rommetjes dat in de cartridgepoort aan de zijkant van de ST moet worden gestoken.
Natuurlijk is het niet zo heel erg simpel als het hier voorgesteld wordt, want de Mac en de ST blijken niet eens zo heel veel hetzelfde te zijn.
Ten eerste heeft de Mac een andere geheugenindeling, waardoor sommige programma's op de ST vastlopen als ze niet bestaand geheugen aanspreken.
Verder verschillen de diskdrives van de Mac en de ST nogal, waardoor het normaal gesproken niet mogelijk is om Macintosh diskettes te lezen. (Hoewel de nieuwste Magic Sac dat met wat bijkomende hardware wel schijnt te kunnen!) Als laatste is de meeste Macintosh software geschreven voor de Apple Imagewriter, waardoor ST-bezitters met meestal Epson kompatibele printers niet veel beginnen.

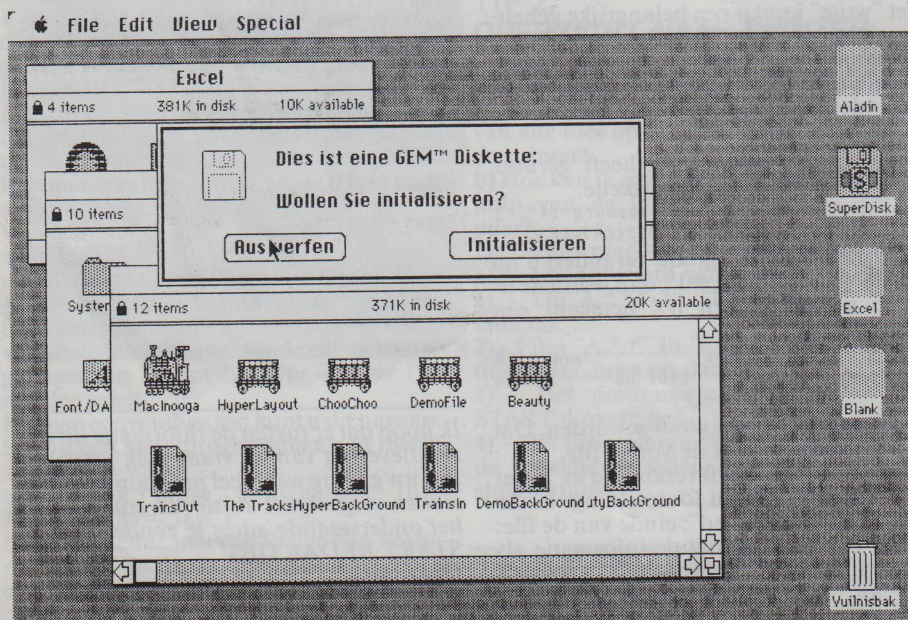
**Aladin**

**Welcome to Macintosh**

©

## SUPERDISK

De mate waarin de makers van de emulator erin slagen om deze verschillen op te lossen, staat voor de bruikbaarheid van hun produkt. De eerste Mac-emulator was de Magic Sac. Deze werd alom als een speeltje opgevat. Om de vijf minuten kreeg de gebruiker een foutmelding met allerlei onbegrijpelijke informatie over het scherm rollen. Het enige wat men wel begreep, was de zin 'Press RETURN or move mouse to restart cartridge'. Later kwamen er betere versies van de Magic Sac uit, maar een echt goed bruikbaar niveau - voor zover ik heb kunnen testen - heeft dit produkt nog niet bereikt.
Voor mij was het Macintosh avontuur afgelopen. Totdat iemand mij op een clubavond in het oor fluisterde: 'Die daar heeft Aladin!!'. Aangenaam verrast stapte ik op de man toe, maakte een babbeltje en kreeg hem zover dat hij een kleine demonstratie gaf.
Nadat de cartridge in de zijkant van de ST was gestoken, werd ik overrompeld door een schitterend grafisch intro. Na verloop van tijd waagde iemand het om tweemaal op de spatiebalk te drukken. 'Aladin, der Macintosh-Enhancer, wird geladen...' Aladin diskette in de drive. 'Welcome to Macintosh', verscheen keurig in het midden van het scherm.
Niet lang daarna prijkte er uiteraard een Aladin bij mij thuis: een mooie witte videodoos met Duitstalige handleiding, een printje en drie diskettes. Printje in de cartridgepoort en aan de slag!
De Macintosh drives hebben geen ejectknop, maar de diskettes moeten softwarematig uit de drives gehaald worden. Als men in de Finder (de desktop van de Mac) op 'Control E' drukt, spuwt de drive automatisch de diskette eruit. Bij Aladin gaat het diskdrive lampje knipperen en wordt er een onderbroken pieptoontje ten gehore gebracht als teken dat de diskette uit de drive mag worden gehaald. Heel aangenaam. Verdere opties in Aladin zijn onder andere het formatteren (op de Mac 'initialiseren' genaamd) vanuit elk programma (zowel enkel- als dubbelzijdig), het herkennen en lezen (niet schrijven) van Magic Sac diskettes, de beschaafde foutmeldingen ('A system error occurred'), de waarschuwende pieptoontjes enz...
De vervelende eigenschappen van de Macintosh worden tot een minimum gereduceerd. Bij de oude Mac moest men steeds bij het opstarten van de Macintosh een gedeelte van de systeemsoftware naladen (vgl. TOS op schijf). Aladin heeft echter een ingebouwde resetbestendige ramdisk (SuperDisk) waarin de systeemsoftware kan worden gekopieerd. Vanaf dat moment is de opstartschijf niet meer nodig.
Niet alleen de ramdisk, maar het hele Aladin systeem is reset-bestendig. Na een druk op de reset-knop krijgt de gebruiker weer het lachende Macintoshje te zien. Is de systeemsoftware al in de SuperDisk gekopieerd, dan ben je binnen een paar sekonden in de Finder. Om weer terug te keren naar TOS moet de ST dus eerst even worden uitgezet.




## SOPHISTICATED

Verreweg het grootste voordeel van Aladin is dat de gebruiker zijn Epson, Nec Pinwriter of een daarmee kompatibele printer kan blijven gebruiken. Aladin konverteert namelijk de Imagewriter stuurkodes in Epson of P6 stuurkodes. Dit betekent niet alleen dat er een

# Space R&D

Reporting On Innovation In The Commercial Space Market

*NASA Interview*

## JAMES BEGGS SPEAKS OUT ON THE PRIVATE SECTOR

*James M. Beggs was appointed Administrator of the American Space Agency in June of 1981. A strong supporter of private sector non-government programs. What follows are excerpted highlights of a mid morning interview with Mr. Beggs.*

pushing too hard for the space station program. What is the game plan for selling the idea to both Congress and the Administration?

**Beggs**: The game plan is to do what we have done at this agency as far as new programs



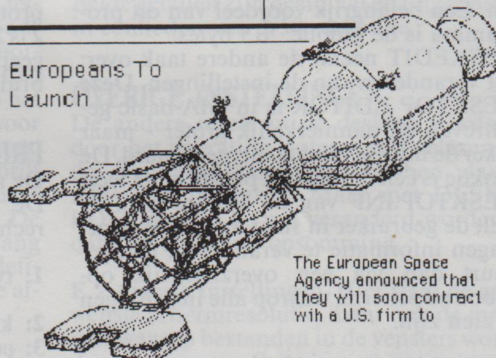
NASA Administrator James M. Beggs

President and the folks at NASA were searching for a program that could not be matched by

### Survey Finds Japan Lacking In Basic Creativity Research

A survey taken of Japanese executives confirms their belief that Japan still lags behind the other major industrial nations in the capacity to develop technological products for the marketplace. The conclusions were reported by the Japanese Council for Science and Technology, and is summarized in the report on "Basic Policies for Long-Term Promotion of Science and Technology." The executives were asked to cite the reasons why Japan remains behind European and American

### Europeans To Launch



The European Space Agency announced that they will soon contract with a U.S. firm to

**START**

Dit is een test van het programma pagemaker op een Atari ST.
Zo te zien werkt het pakket uitstekend!
Dat is natuurlijk een grote verdienste van de makers van Aladin...
Aan de andere kant wordt meteen duidelijk hoe intensief publishing partner leentjebuur heeft gespeeld.
Je kunt het pakket bijna met dezelfde toetsen gebruiken.

briefje van WriteNow kan worden uitgeprint, maar ook dat de ST bezitter in staat is om echt zeer mooie Desktop Publishing programma's als 'Ready Set Go' en 'Pagemaker' te gebruiken. Natuurlijk: niemand hoeft met de Desktop Publishing rage mee te doen, maar pak dan maar eens het krachtige geïntegreerde database/spreadsheet-pakket Excel (voor het geld moesten we het tot nu toe laten).
Een voordeel dat niet zozeer aan Aladin maar aan de ontwerpers van de ST is te danken, is het feit dat de Mac-emulator ongeveer 20% sneller loopt dan de oude Mac (we hebben het eventjes niet over de Mac II) en dat de beeldschermresolutie hoger is.
Maar alles heeft zijn nadelen. Ook Aladin. Het grootste nadeel is wel dat niet alle software probleemloos loopt. Vaak is het nodig om de programma's zodanig te veranderen ('patchen'), dat deze onder Aladin ook goed werken. Bij de Aladin-versie van juni '87 wordt dit patchen echter tot een minimum beperkt. Niettemin zal er nog altijd software zijn die niet goed onder Aladin werkt. Een deel van de Macintosh software valt dus af.
Een tweede nadeel wordt gevormd door de handleiding. Deze is specifiek gericht op Macintosh gebruikers en iemand die nog nooit intensief op een Macintosh heeft gewerkt moet zijn informatie dus ergens anders vandaan halen. (Tip: 'Het Macintosh gebruikershandboek'). De gebrekkige handleiding valt enigzins te verklaren uit het feit dat Proficomp Aladin als een Macintosh- en niet als een Atari-produkt op de markt brengt. Dit artikel zou hier dus niet geheel en al op z'n plaats zijn. Vermoedelijk is dit gedaan om de kans op een proces van de kant van Apple zo klein mogelijk te houden.
Om Macintosh software onder Aladin te kunnen draaien, moet de gebruiker deze software op Aladin of Magic Sac formaat op schijf hebben staan. Anders is het noodzakelijk een ST via de RS232-poort aan een Mac te koppelen om op die manier aan de benodigde software te komen. Daarvoor is dan wel eventjes een Macintosh nodig - en waar haal je die nu zomaar vandaan? En als dat uiteindelijk lukt, blijkt dat het overzenden een langdurige en vervelende bezigheid is.
Tenslotte kan men zich afvragen of er wel behoefte is aan een Macintosh-emulator. Een goede vriend van mij zegt altijd dat de Macintosh software zo 'sophisticated' is. Wie wel eens een documentje met WriteNow heeft gemaakt, weet wat hij bedoelt. Geen hinderlijk ge'reformat' (vgl. 1ST Word), een flink aantal fonts, geheel proportioneel en snel. Inderdaad lijken de Macintosh programma's (die over het algemeen duurder zijn dan hun ST-equivalenten) een hogere graad van verfijning te hebben bereikt. Misschien is dat een reden tot aanschaf van Aladin.

*Egon Lubbers*

*Aladin zonder Macintosh ROMs kost rond de 350 gulden. Een kastje om het printje kon er voor die prijs helaas niet vanaf. Voor de noodzakelijke ROMs moet er nog zo'n 200 gulden worden bijgeteld.*
*Aladin ondersteunt ST machines met TOS in ROM en een RAM tot 2.5 Mb. Verder zijn er printerdrivers voor Epson en Nec P6 kompatibelen. Een monochroom beeldscherm is natuurlijk een vereiste.*
*Het pakket wordt in Nederland geimporteerd door Softpack.*

File Edit View Format Font FontSize Style

Venice
✓Sydney
Stuart
Saigon
Richardson
Rangers
London
Florence
Cursive
Carmel
Camelot
Boise
Helvetica
Athens
Chicago
Geneva
New York
Monaco

titled-2

DE MUZI ANT VAN DE ST

Dit artikel is be u een beetje vertrouwd te maken met de enerator van de ST. Atari koos hiervoor d van Yamaha. Dit is niet de mooiste gelui je je kunt voorstellen (in de ZX81 zit er he en), maar er zijn wel heel aardige resultat bereiken.
Om plezier van d te hebben moet u in het bezit zijn van ee ler of GFA Basic.

Het TOS bevat een geluidsverwerking tot een fluitje van een cen bios(31). Deze functie heeft alleen een adres n adres moet de functie de data voor het af te spelen geluid aantreffen. Vervolgens is de gebruiker verplicht te wachten totdat de ST klaar is met het spelen van z'n riedeltje.
Wie zelf zo'n liedje wil schrijven, moet weten in welk formaat de functie bios(31) de data verlangt. In

Page 1

# DE DESKTOP ONDER DE LOEP

**Er zullen maar weinig ATARI-gebruikers zijn die het bestand DESKTOP.INF nog nooit zijn tegengekomen. Het wordt namelijk aangemaakt als vanaf de desktop het menu-item Save Desktop gekozen wordt. DESKTOP.INF bevat de "huidige" instellingen van een aantal zaken - onder andere de kleur-definities en de posities van de vensters op de desktop. Maar ook de instellingen van de printer- en modempoort staan in deze file opgeslagen. Wie die instellingen wil wijzigen, heeft de desktop-accessoires nodig. Maar die nemen een fors stuk geheugen en vier plaatsen in de menubalk in beslag. Bovendien vragen ze nogal wat laadtijd. U begrijpt het al: START heeft een oplossing voor u.**

## VERSPILLING

Telkens wanneer de computer opstart (na een koude start, of na een reset) leest de ST het bestand DESKTOP.INF en neemt hij de gevonden instellingen over. Althans: de meeste. Voor bepaalde instellingen, zoals voor de poorten, zijn aparte accessoires nodig: het bekende duo CONTROL.ACC en EMULATOR.ACC. Het controlpanel stelt de kleuren in, de snelheid van muis en toetsenbord, de tijd, de datum en de printerpoort. EMULATOR.ACC zorgt onder andere voor de instelling van de seriële poort. Deze instellingen kunnen via de accessoires veranderd worden met de respektievelijke menu-titels "Install Printer" en "Set RS232 Config."

Normaal gesproken werkt dit feilloos: de desktop zorgt voor een gedeelte van de instellingen en de accessoires doen de rest. Er is maar één nadeel: de accessoires moeten op elke opstartschijf staan. Dat zou zo erg nog niet zijn, als ze niet elk TWEE menu-titels in beslag namen! En die zijn nu eenmaal schaars: er zijn er maar zes te verdelen tussen ALLE accessoires. Als CONTROL.ACC en EMULATOR.ACC geladen zijn, is er dus nog maar ruimte voor twee andere accessoires... Bovendien worden de standaard-accessoires meestal niet meer gebruikt als de juiste instellingen eenmaal gevonden zijn. Helaas moeten ze zelfs dan op de bootdisk blijven, omdat ATARI er bijvoorbeeld vanuit gaat dat de printer standaard met 1280 dots per inch werkt, terwijl dat voor de wijdverspreide EPSON-printers toch echt 960 moet zijn. Om nog maar niet te spreken van gebruikers van een seriële printer. Die moeten de RS232-poort instellen en de modempoort als printerpoort gebruiken (niet de parallelle Centronics-poort).

Maar ook het veranderen van de instellingen is vaak een beetje moeizaam: accessoires aktief maken, computer resetten, instellen, uitproberen, accessoires deaktiveren. Met een harddisk valt dat allemaal nog wel mee, maar met floppies kan het behoorlijk vervelend werken. Dat moet toch anders kunnen...

## 515 BYTES!

De standaard accessoires hebben dus twee taken: het veranderen van de desktop-instellingen en het instellen zelf. Om ze overbodig te maken, moeten deze twee taken dus op een andere manier vervuld worden. Met behulp van de twee programma's die we u hier presenteren is dat inderdaad mogelijk: SET-DESK en DESK-EDIT.

Het eerste is een klein assembler-programma, dat bedoeld is om vanuit de AUTO-folder te werken. Het onderzoekt of het DESKTOP.INF-bestand zich op de huidige disk bevindt. Zo ja, dan haalt het de informatie op en voert verschillende instellingen uit, onder andere de beide poorten. Een belangrijk voordeel van dit programma is de grootte: 515 bytes!

DESKEDIT neemt de andere taak over: het veranderen van de instellingen. Deze DESKTOP EDITOR is in GfA-basic geschreven en aanmerkelijk groter - maar zeker de moeite van het intypen waard. De funktie is eenvoudig: het programma leest DESKTOP.INF van de huidige disk en stelt de gebruiker in staat de daarin opgeslagen informatie te veranderen. Dat gebeurt dan via een overzichtelijk opgebouwd scherm, waarop alle instellingen te zien zijn.

## HEKJES

Voordat de beide programma's besproken worden, eerst nog wat informatie over de opbouw van het DESKTOP.INF-bestand.

Het DESKTOP.INF-bestand is een gewone tekst-file. Dat wil zeggen: hij bestaat uit tekstregels en kan dus direkt op het scherm of op de printer worden afgedrukt. Bekijk hem maar eens door twee keer snel op de naam te klikken en dan te kiezen voor Show. Op het eerste gezicht is er geen touw aan vast te knopen, maar het is meteen duidelijk dat elke regel met een hekje begint. Daarachter staat een letter, die de betekenis van die regel aangeeft. Voor de instellingen die normaal via de accessoires worden verzorgd, zijn er maar drie van belang: de regels waarvan het tweede karakter (na het hekje dus) een a, een b of een c is. De regel met de a geeft de instelling van de seriële poort aan; letter b is voor de printerpoort en de c staat voor de scherminstellingen. De betekenis van de andere regels komt verderop kort aan de beurt.

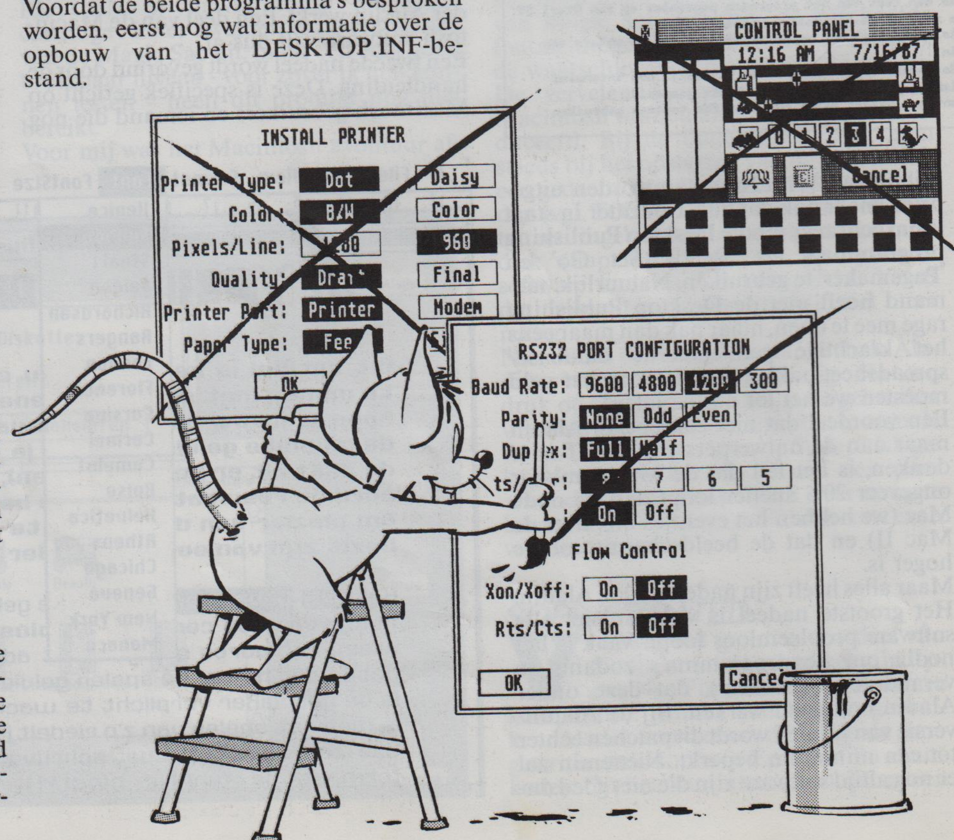
## MODEMPOORT

Om met de a te beginnen: hierachter staan zes cijfers, die elk een aparte betekenis voor de modem-poort hebben. Het eerste staat voor full duplex (1) of half duplex (0); cijfer twee bepaalt de snelheid van de poort in baud: 0 is 9600, 1 is 4800, 2 is 1200, 3 is 300. Het derde geeft de pariteit aan: 0 is geen, 1 is oneven, 2 is even. Daarachter volgt het aantal bits per karakter: 0 betekent 8, 1 betekent 7, enzovoort. Het een-na-laatste cijfer staat voor het gebruikte protocol: 0 is geen protocol, 1 is RTS/CTS, 2 is Xon/Xoff, 3 is allebei. Het laatste cijfer geeft aan, of bit 8 (het meest linkse bit) gebruikt wordt. 0 betekent nee, 1 is ja.

## PRINTERPOORT

De regel met de b stelt de printerpoort in. De zes cijfers hebben (van links naar rechts) de volgende betekenis:

| | | |
|---|---|---|
| **1:** | type | 0 is matrix, 1 is margrietwiel. |
| **2:** | kleur | 0 is nee (zwart/wit), 1 is ja. |
| **3:** | puntdichtheid | 0 is 1280, 1 is 960 puntjes per inch. |



DESKTOP EDITOR Version 1.4 © 01-05-1987 By R.van der Linden

RS232 (#a)
Full Duplex
300 Baudrate
None Parity
8 Bits/char
None Protocol
On Stripbit

Printerport (#b)
Dot Type
B/W Color
960 Pixels/line
Draft Quality
Printer Port
Feed Papertype

Control (#c)
3 Mouse_response
On Keyboardclick
On Bell
11 Keyboard_pause
3 Keyboard_response
White Bordercolor

Icon Screen
D D D T

Set Application
Read Write Reset
Exit

Menu_bar (#E)
High Resolution
On Confirm Copy's
On Confirm Deletes
Type Sort by
Icon I/T Select

Window Screen

**4:** kwaliteit 0 is test, 1 is definitief.
**5:** printer- poort 0 is de parallelle, 1 is de seriële poort.
**6:** papier- soort 0 is kettingformulieren, 1 is losse vellen.

In de praktijk is vooral de puntdichtheid van belang: dit moet voor alle EPSON-compatibele printers 960 zijn, omdat anders een afdruk van het scherm (bijvoorbeeld na het indrukken van ALT-HELP) te breed is en horizontaal niet meer op het papier past. Maar ook de poort zelf is voor gebruik van een seriële printer van belang. De instelling van de papiersoort wordt vaak door tekstverwerkers bekeken, om al of niet tussen de pagina's te pauseren. De kwaliteit tenslotte, is ook weer van belang voor onder andere screendumps: de "definitieve" kwaliteit geeft een wat zwartere afdruk, die dan ook wat langer duurt.

## SCHERM

Tenslotte de letter c. Deze regel bevat een heleboel verschillende informatie. Na de c volgen eerst zestien groepen van drie cijfers. Dit zijn de instellingen van de zestien kleuren: de drie cijfers geven respektievelijk de rood-, groen- en blauwcomponent van die kleur aan. Deze zijn met het controlpanel ingesteld: ze worden dan ook alleen "geladen" als CONTROL.ACC op de bootdisk staat. (Let op: de volgorde van de kleuren is niet helemaal logisch!).
Na de 48 kleurencijfers volgen nog zeven andere getallen. Het eerste cijfer daarvan heeft een waarde tussen 0 en 4 en geeft de muissnelheid aan. Nummer twee is een 0 of een 1: voor toetsklik uit en aan. Het derde cijfer is een 0 als de "ping" uit staat, anders een 1. Als de ping aan staat, wil dat zeggen dat het belsignaal mogelijk is. Dit is ook uit te zetten en dan is het helemaal niet meer mogelijk een bel te laten klinken.
De laatste vier cijfers vormen twee groepen van twee, die iets over de instelling van het toetsenbord zeggen. Er kunnen namelijk twee dingen veranderd worden aan de manier waarop het toetsenbord karakters herhaalt. In het controlpanel wordt dit grafisch weergegeven met twee balken: de eerste geeft aan hoe diep een toets moet worden ingedrukt, de andere heeft als uitersten een schildpad en een muis. De betekenis van die bovenste balk is niet helemaal duidelijk uit het plaatje. Het gaat erom hoe lang een toets moet worden ingedrukt voordat het herhalen begint. Dit is ook de betekenis van het eerste tweecijferige getal in de c-regel. De balk geeft aan hoe snel een toets herhaalt, oftewel hoe lang de tijd tussen twee herhalingen is. Dit staat in de c-regel in de laatste twee kolommen. Beide getallen zijn decimaal en uitgedrukt in eenheden van 1/50e sekonde.

## OVERIGE INSTELLINGEN

De andere regels in de desktop worden door het operating systeem zelf gelezen - de standaard-accessoires hebben daar niets mee te maken. Maar met DESKEDIT kunnen ze wel veranderd worden: daarom een korte opsomming.

**E** - diverse instellingen. Hier staat de gewenste schermresolutie, maar ook de manier waarop bestanden in de vensters worden weergegeven (het view- en optionsmenu in de desktop).
**W** - de ligging en inhoud van maximaal vier vensters die meteen geopend moeten worden als de computer opstart.
**M** - ikonen van de floppies. Naam en letter van de disk, evenals de positie (en de vorm!) van het ikoon.
**T** - positie en ikoon van de vuilnisbak.
**G** - als GEM-applicatie aangemelde programma's.
**F** - idem, met TOS-applicaties.
**P** - Tos-Takes-Parameters TTP-applicaties.

## SET-DESK

Dit programma is twee keer afgedrukt; eenmaal als assembler-listing en in de vorm van DATA-regels in GfA-Basic. Het laatste programma maakt een file SET-DESK.PRG, die 515 bytes lang is. Het programma wordt voorafgegaan door een checksum-programma. Het is handig om dit apart te bewaren, daar START er in de toekomst vaker van gebruik hoopt te maken.
In een notedop werkt SET-DESK als volgt: het bepaalt eerst of de file DESKTOP.INF op de huidige disk staat. Als dat zo is, filtert SET-DESK de relevante informatie eruit en verzorgt de instellingen van de modempoort, de printerpoort, het toetsenbord en de kleuren op het beeldscherm - alle instellingen, die normaal door de standaard accessoires worden verricht. Alleen de muissnelheid en het stripbit voor de RS232 worden genegeerd.
Samenvattend: SET-DESK handelt de regels #a, #b en #c uit DESKTOP.INF af.

## XBIOS-FUNKTIES

Voor ingewijden: het eigenlijke instellen van zowel de beide poorten als de kleuren gebeurt met zogenaamde Xbios-calls. Voor de kleuren de funktie Setcolor (nummer 7), voor de seriële poort nummer 15, Rsconf, en voor de printer funktie nummer 33, Setprt. De gegevens voor het toetsenbord worden ingesteld met Xbios-funktie 35: kbrate. Op de parameters voor deze funkties gaan we hier nu niet in. Het aan- of uitzetten van de toetsklik en de ping gebeurt door een bit in de systeemvariabele Conterm te zetten.

## DESKEDIT

Om te beginnen een paar punten om op te letten. DESKEDIT werkt alleen goed in samenwerking met een monochrome monitor. Verder: maak altijd eerst een kopie van de oude DESKTOP.INF, alvorens hem te veranderen. DESKEDIT overschrijft namelijk de oude versie. Als er onverhoopt iets fout mocht gaan, kan altijd de kopie van stal gehaald worden.
Als de editor wordt opgestart, verschijnt er een grafische weergave van alle instellingen van de desktop. De informatie is in blokken gegroepeerd. Bovenin het scherm staan van links naar rechts de instellingen van de seriële poort, de printerpoort en het scherm. Deze corresponderen met de instellingen die normaal gesproken door de standaard accessoires verricht worden, maar die dus ook overgenomen kunnen worden door SET-DESK: #a, #b en #c.
Rechts in het midden bevindt zich nog zo'n blok. Hierin staan de instellingen die via het desktopmenu toegankelijk zijn: het bevestigen van kopiëren en verwijderen van bestanden en de schermresolutie. Verder staat hier hoe bestanden in de vensters moeten worden afgebeeld: als ikonen of als tekst en hoe ze gesorteerd moeten worden (op naam, type, enzovoort). Deze instellingen zijn een onderdeel van de regel met #E in DESKTOP.INF.
In elk van deze blokken staan de huidige instellingen. Ze kunnen veranderd worden door met de linker muisknop op de pijltjes te klikken. De pijltjes geven de "richting" van de verandering aan. Klikken met beide muistoetsen tegelijk geeft een repeterend effekt; vooral handig bij control-instellingen. Na veranderen verschijnen de nieuwe instellingen meteen op het scherm.

## IKONEN INSTALLEREN

Links van het midden staat het ikoonscherm. Dit is een schematische weergave van het beeldscherm. Op de plaatsen waar een D staat, komt in werkelijkheid een disk-ikoon; waar een T staat komt een vuilnisemmer. Ook dit deel van de desktop editor wordt met de muis bestuurd. Klikken in een leeg vakje levert een soort dialoogscherm op, waarin een nieuw ikoon geïnstalleerd kan worden. Als er in een bezet vakje geklikt wordt, kan het bij dat vak behorende ikoon gewijzigd worden. Het is, net als op de desktop, niet mogelijk meer ikonen over elkaar heen te zetten.
In het dialoogscherm kunnen de instellingen veranderd worden door op de venstertekentjes te klikken. De linker muistoets werkt hier tegengesteld aan de rechter.

Natuurlijk zijn ikonen ook gewoon via de desktop te installeren (met Install diskdrive onder andere), maar DESKEDIT voegt wat extra mogelijkheden toe. Zo kunnen er meer vuilnisbakken worden geïnstalleerd, bijvoorbeeld één in elke hoek van het scherm. Maar ook de vorm van alle ikonen kan gekozen worden: elk kan de vorm van een programma-, folder-, disk- of vuilnisemmerikoon krijgen. Dit kan hele onoverzichtelijke desktops tot gevolg hebben: vuilnisemmers vermomd als diskdrives bijvoorbeeld. Maar het kan juist weer heel handig zijn een RAM-disk de vorm van een folder te geven. Met DESKEDIT is dat alles in een vloek en een zucht in te stellen. Deze instellingen worden in DESKTOP.INF opgeslagen in de regels #M (voor disks) en #T voor vuinisemmers.

## VENSTERS EDITTEN

Het laatste gedeelte is het vensterscherm, dat het onderste deel van het beeld in beslag neemt. Ook dit is een schematische weergave van het echte scherm, met daarop de vensters die bij het opstarten geopend worden. Klik ergens in het scherm en er verschijnt weer een dialoogscherm, waarin de gebruiker de posities en groottes van de vier mogelijke vensters kan opgeven. Hier voegt DESKEDIT weer wat toe: bij elk venster kan ook worden aangegeven welke files erin moeten worden afgebeeld! Het kan van pas komen alleen de files *.PRG in de folder A:/NIX/NOGGES te zien bij het opstarten. In dat geval moet de padnaam van een van de vensters op
A:/NIX/NOGGES/*.PRG
gezet worden. Dat kan met DESKEDIT in de vensterdialoog. Helaas is dit gemak maar van korte duur - bij het opstarten wordt het venster keurig geopend en inderdaad: alleen de files met extensie PRG zijn erin te zien. Maar wanneer er een andere folder open gaat, of na het sluiten en weer openen van het venster, zijn toch alle bestanden weer zichtbaar.
De gegevens van al deze vensters worden in DESKTOP.INF opgeslagen in de regels die met #W beginnen.

## KNOPPEN

Tenslotte staan er nog vijf knoppen in het midden van het scherm. Ze kunnen worden "ingedrukt" door er met de muis op te klikken.
**READ** leest DESKTOP.INF van de huidige disk en laat alle instellingen op het scherm zien. DESKEDIT moet dus altijd worden opgestart vanaf de disk van waar DESKTOP.INF gelezen moet worden.
**WRITE** schrijft de instellingen weg naar DESKTOP.INF; ook weer op de huidige disk. De oude versie wordt overschreven door de nieuwe.
**RESET** maakt alle instellingen ongedaan. Het scherm geeft een reeks instellingen te zien die in praktijk vaak voorkomt. Deze zijn in DESKEDIT ingebouwd.
**EXIT** beeindigt het programma. Zorg ervoor dat de instellingen met WRITE zijn weggeschreven, anders was al het werk voor niets.
**SET APPLICATION** tenslotte, doet hetzelfde als Install Application op de desktop. De instellingen verlopen ook op ongeveer dezelfde manier; eventueel aanwezige instellingen (uit de regels #G, #F en #P van DESKTOP.INF) kunnen ook veranderd worden. Het is mogelijk aan programma's die bij een bepaalde applicatie horen een eigen ikoon mee te geven. Op deze manier kunnen GfA-programma's er bijvoorbeeld uitzien als folders.

## OVERBODIG?

Zijn de standaard-accessoires CONTROL.ACC en EMULATOR.ACC hiermee nu overbodig geworden? Bijna, maar nog niet helemaal. Een aantal instellingen wordt wel door de accessoires, maar niet door SET-DESK verricht (de muissnelheid en het stripbit). En DESKEDIT kan de schermkleuren niet veranderen. Bovendien zorgt het controlpanel ervoor, dat de tijd na een reset bewaard blijft en de tijd en datum kunnen worden ingesteld. Hier kunnen SET-DESK en DESKEDIT dus samen minder dan de accessoires.
Maar de voordelen hebben de overhand: SET-DESK is bij het opstarten vele keren sneller dan de accessoires. En hoe vaak heeft die wachttijd u niet al geërgerd? Bovendien legt het programma nauwelijks beslag op het kostbare geheugen van uw ST. Maar het belangrijkste voordeel is misschien wel dat de volle zes menutitels voor accessoires beschikbaar zijn, zonder dat de helft van de instellingen in DESKTOP.INF verloren gaat.
DESKEDIT kan een paar zaken instellen die met het controlpanel niet mogelijk zijn. Meer vuilnisbakken bijvoorbeeld, of padnamen in de vensters. Bovendien is DESKEDIT een onafhankelijk programma. De gebruiker kan het op elk gewenst moment aktiveren. Wie de desktop met de standaard accessoires wil bewerken, moet deze op de bootschijf kopiëren, daarna de machine resetten en, na gedane arbeid, CONTROL en EMULATOR weer wissen en nogmaals resetten.
SET-DESK en DESKEDIT vullen elkaar zo goed aan dat we er zeker van zijn dat ze onafscheidelijke stukken gereedschap voor u zullen worden.

*Marcus The (tekst en SET-DESK)*
*René van der Linden (DESKEDIT)*

● *LISTING 1*

```
' CHECKSUM door Marcus The
' een programma om binaire files
' te maken uit data-statements
' formaat:
' Data Filenaam, Aantal bytes
' Data [Groep van 8 hex bytes],Checksum
'
Read F$
Read L%
Clr T%
Open "o",#1,F$
Print "File: ";F$
Print "Totaal: ";L%;" bytes."
Print
For I%=1 To L% Step 8
  Print Chr$(13),I%,;
  Clr S%
  For J%=1 To 8
    Read A$
    A%=Val("&h"+A$)
    Add S%,J%*A%
    Print #1,Chr$(A%);
    Inc T%
    Exit If T%=L%
  Next J%
  Read S2%
  If S2%<>S%
    A$="Fout in regel "+Str$(((I%-1)/8)+1)
    Alert 3,A$,1," Verder | Stoppen ",J%
    If J%=2
      Close
      End
    Endif
  Endif
Next I%
End
'
' Hierachter moeten de DATA-regels komen,
' hetzij door MERGEn, hetzij door typen
' BEWAAR DE BOVENSTAANDE ROUTINE
'
Data set_desk.prg,515
' Data-regel 1
Data 60,1A,00,00,01,E6,00,00,1533
Data 00,00,00,00,00,00,00,00,0
Data 00,00,00,00,00,00,00,00,0
Data 00,00,FF,FF,48,7A,01,E4,4708
Data 3F,3C,00,1A,4E,41,5C,8F,2855
Data 3F,3C,00,00,48,7A,01,A6,2610
Data 3F,3C,00,4E,4E,41,50,8F,2979
Data 4A,40,6B,42,3F,3C,00,00,1462
Data 48,7A,01,92,3F,3C,00,3D,2066
Data 4E,41,50,8F,4A,40,6B,2E,2891
' Data-regel 11
Data 3C,00,48,7A,01,AE,2F,3A,2606
Data 01,C4,3F,06,3F,3C,00,3F,1785
Data 4E,41,DF,FC,00,00,00,0C,1981
Data 3F,06,3F,3C,00,3E,4E,41,1942
Data 58,8F,48,7A,00,10,3F,3C,2095
Data 00,26,4E,4E,5C,8F,3F,3C,2861
Data 00,00,4E,41,41,FA,01,7C,3318
Data 42,43,0C,10,00,1A,67,14,1337
Data 0C,18,00,23,67,10,0C,10,1023
Data 00,1A,67,08,0C,18,00,0A,677
' Data-regel 21
Data 66,F4,60,E6,4E,75,0C,10,3102
Data 00,61,66,56,54,88,42,40,3054
Data 61,00,01,0A,43,FA,01,2C,2334
Data 10,31,30,00,61,00,00,FE,2775
Data 34,3C,00,98,43,FA,01,20,2878
Data D4,31,30,00,61,00,00,EE,2843
Data 43,FA,01,18,D4,31,30,00,2356
Data 42,43,61,00,00,E0,2F,08,2228
Data 3F,3C,FF,FF,3F,3C,FF,FF,6468
Data 3F,3C,FF,FF,3F,02,3F,03,2760
' Data-regel 31
Data 3F,00,3F,3C,00,0F,4E,4E,1752
Data DF,FC,00,00,00,0E,20,5F,1795
Data 60,94,0C,10,00,62,66,2E,2162
Data 52,88,3E,3C,00,05,43,FA,3279
Data 00,DC,42,40,16,18,B6,3C,2902
Data 00,30,67,04,D0,31,70,00,2539
Data 51,CF,FF,F2,2F,08,3F,00,2952
Data 3F,3C,00,21,4E,4E,58,8F,2933
Data 20,5F,60,00,FF,62,0C,10,2585
Data 00,63,66,00,FF,5A,52,88,3981
' Data-regel 41
Data 43,FA,00,B0,2E,3C,00,00,1861
Data 00,0F,78,00,18,18,04,04,714
Data 00,30,E1,44,42,43,61,64,3254
Data E9,43,88,43,42,43,61,5C,3190
Data 88,43,42,40,10,19,48,E7,3306
Data 80,C0,3F,04,3F,00,3F,3C,1953
Data 00,07,4E,4E,5C,8F,4C,DF,4194
Data 03,01,51,CF,FF,CE,41,E8,5898
Data 00,01,14,39,00,00,04,84,1374
Data 08,C2,00,00,61,2E,66,04,1903
' Data-regel 51
Data 08,82,00,00,08,C2,00,02,1488
Data 61,22,66,04,08,82,00,02,1323
Data 13,C2,00,00,04,84,61,1C,2122
Data 34,03,61,18,3F,03,3F,02,1235
Data 3F,3C,00,23,4E,4E,5C,8F,2969
Data 60,00,FE,DC,16,18,04,03,2044
Data 00,30,4E,75,42,43,61,F4,4161
Data C6,FC,00,0A,38,03,61,EC,3607
Data D6,04,4E,75,5C,44,45,53,2939
Data 4B,54,4F,50,2E,49,4E,46,2574
' Data-regel 61
Data 00,00,01,02,07,09,00,04,132
Data 06,00,00,20,20,10,08,04,478
Data 02,01,00,0F,01,02,04,06,157
Data 03,05,07,08,09,0A,0C,0E,367
Data 0B,0D,00,37
```

● *LISTING 2*

```
*       Set-desk door Markus The
*       een programma om DESKTOP.INF te lezen en "uit te voeren"

*       lees DESKTOP.INF:
        pea     buffer(pc)
        move.w  #$1A,-(sp)
        trap    #1
        addq.l  #6,sp           ;zet transfer adres op buffer
        move.w  #0,-(sp)        ;attribute: normale file
        pea     naam(pc)        ;zoek naar \DESKTOP.INF
        move.w  #$4E,-(sp)      ;search first
        trap    #1
        addq.l  #8,sp
        tst.w   d0              ;gevonden?
        bmi.s   ho              ;nee. terug naar TOS.
        move.w  #0,-(sp)        ;ja, gevonden
        pea     naam(pc)
        move.w  #$3D,-(sp)
        trap    #1
        addq.l  #8,sp           ;open \DESKTOP.INF
        tst.w   d0
        bmi.s   ho              ;fout? dan stoppen
        move.w  d0,d6           ;berg file-handle op in d6
        pea     buffer(pc)
        move.l  buffer+26(pc),-(sp)
        move.w  d6,-(sp)
        move.w  #$3F,-(sp)      ;lees hele file naar buffer
        trap    #1
        add.l   #12,sp
        move.w  d6,-(sp)
        move.w  #$3E,-(sp)
        trap    #1
        addq.l  #4,sp           ;sluit de file weer
        pea.l   doit(pc)
        move.w  #38,-(sp)
        trap    #14             ;voer doit uit in supervisor mode
        addq.l  #6,sp
ho      move.w  #0,-(sp)
        trap    #1              ;exit naar TOS

*       de eigenlijke installatie-routine:
doit    lea.l   buffer(pc),a0   ;a0 wijst naar buffer
        clr.w   d3
ishek   cmp.b   #26,(a0)        ;EOF?
        beq.s   einde
        cmp.b   #'#',(a0)+      ;is het een hekje?
        beq.s   hekje
islf    cmp.b   #26,(a0)        ;is het een EOF?
        beq.s   einde
        cmp.b   #10,(a0)+
        bne.s   islf            ;sla karakters over tot LF of EOF
        bra.s   ishek
einde   rts

*       de verschillende installaties: de RS232-poort
hekje   cmp.b   #'a',(a0)       ;RS232-poort?
        bne.s   centr           ;nee. probeer centronics-poort
        addq.l  #2,a0           ;verhoog a0 (sla eerste letter over!)
        clr.w   d0              ;dit wordt de baudrate
        bsr     getch           ;haal volgende letter
        lea     bauds(pc),a1
        move.b  0(a1,d3.w),d0   ;haal baudrate uit tabel
        bsr     getch
        move.w  #152,d2         ;pariteit en lengte
        lea     parit(pc),a1
        add.b   0(a1,d3.w),d2
        bsr     getch
        lea     bits(pc),a1     ;woordlengte
        add.b   0(a1,d3.w),d2
        clr.w   d3              ;handshake
        bsr     getch
        move.l  a0,-(sp)        ;bewaar a0
        move.w  #-1,-(sp)
        move.w  #-1,-(sp)
        move.w  #-1,-(sp)
        move.w  d2,-(sp)
        move.w  d3,-(sp)
        move.w  d0,-(sp)
        move.w  #15,-(sp)
        trap    #14
        add.l   #14,sp          ;stel de poort in
        move.l  (sp)+,a0        ;herstel a0
        bra.s   islf            ;ga verder met scannen

*       centronics-poort
centr   cmp.b   #'b',(a0)       ;centronics-poort?
        bne.s   control         ;nee, verder proberen
        addq.l  #1,a0           ;verhoog a0
        move.w  #5,d7           ;doe 6 keer
        lea     printer(pc),a1
        clr.w   d0              ;hier komt het printer-byte
letter  move.b  (a0)+,d3        ;volgende cijfer
        cmp.b   #'0',d3
        beq.s   lett2           ;een 0: doe niets
        add.b   0(a1,d7.w),d0   ;voeg een bit toe
lett2   dbra    d7,letter       ;verder
        move.l  a0,-(sp)        ;bewaar a0
        move.w  d0,-(sp)        ;printer byte
        move.w  #33,-(sp)
        trap    #14
        addq.l  #4,sp           ;stel de printer in
        move.l  (sp)+,a0        ;herstel a0
        bra     islf            ;ga verder met scannen

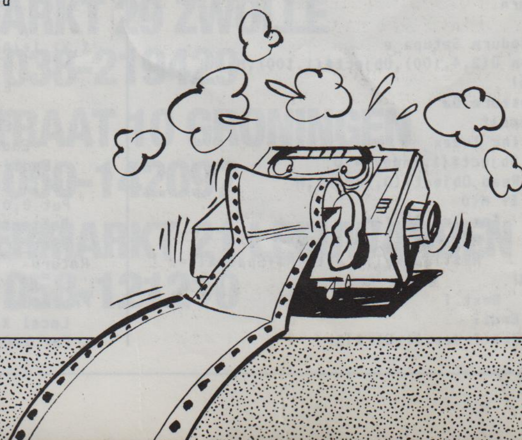
*       de "controle"-regel:
control cmp.b   #'c',(a0)
        bne     islf
        addq.l  #1,a0
        lea     kleurt(pc),a1   ;kleurtabel
        move.l  #15,d7          ;16 kleuren
kleur   moveq   #0,d4           ;stel een waarde samen in d4
        move.b  (a0)+,d4        ;haal het eerste cijfer
        sub.b   #'0',d4
        asl.w   #8,d4           ;8 plaatsen naar links
        clr.w   d3              ;volgende cijfer
        bsr.s   getch
        asl.w   #4,d3           ;4 plaatsen naar links
        or.w    d3,d4
        clr.w   d3
        bsr.s   getch           ;laatste cijfer
        or.w    d3,d4
        clr.w   d0
        move.b  (a1)+,d0
        movem.l a0-a1/d0,-(sp)  ;bewaar a0, a1 en d0
        move.w  d4,-(sp)        ;palette-kleur: 3 cijfers
        move.w  d0,-(sp)        ;kleur-nummer
        move.w  #7,-(sp)        ;setcolor
        trap    #14             ;xbios
        addq.l  #6,sp
        movem.l (sp)+,a0-a1/d0  ;herstel a0, a1, d0
        dbra    d7,kleur        ;al 16 gehad? dan doorgaan
        lea     1(a0),a0        ;verhoog a0 (muis-snelheid overslaan)
        move.b  $484,d2         ;haal CONTERM op
        bset.b  #0,d2           ;neem aan: toetsklik
        bsr.s   getch           ;toets-klik?
        bne.s   tk_ja
        bclr.b  #0,d2           ;nee: bit 0 uit
tk_ja   bset.b  #2,d2
        bsr.s   getch           ;BELL bij ctrl-G?
        bne.s   cg_ja
        bclr.b  #2,d2
cg_ja   move.b  d2,$484         ;nieuwe CONTERM
        bsr.s   get2ch          ;haal twee cijfers naar d3
        move.w  d3,d2           ;dit is de vertraging
        bsr.s   get2ch          ;twee getallen naar d3: herhaaltijd
        move.w  d3,-(sp)        ;herhaaltijd
        move.w  d2,-(sp)        ;vertraging
        move.w  #35,-(sp)       ;kbrate xbios-funktie
        trap    #14
        addq.l  #6,sp
        bra     islf            ;klaar.

*       handige subroutines:
getch   move.b  (a0)+,d3        ;haal een letter uit de file
        sub.b   #'0',d3         ;reken om naar een getal
        rts
get2ch  clr.w   d3              ;haal een twee-cijfer getal naar d3
        bsr.s   getch           ;haal getal naar d3
        mulu    #10,d3          ;maal 10
        move.w  d3,d4
        bsr.s   getch           ;nog eens
        add.b   d4,d3           ;tel op: 2 cijfers in d3
        rts

*       data: de filenaam
naam    dc.b    '\DESKTOP.INF',0

*       tabellen:
bauds   dc.b    1,2,7,9
parit   dc.b    0,4,6,0
bits    dc.b    0,32
printer dc.b    32,16,8,4,2,1
kleurt  dc.b    0,15,1,2,4,6,3,5,7,8,9,10,12,14,11,13
buffer  ds.w    0               ;hier wordt de file ingelezen

        end
```



● LISTING 3

```
Cls
Clear
Defmouse 2
Deftext ,0,0,6
Deffill ,0,0
Dim Invoer$(50),Sp(4,10),Ap$(5,100),M =>
uis(10,4)
Nam$="\DESKTOP.INF"
Fullw 1
Openw 1
Clearw 1
Titlew 1," DESKTOP EDITOR  Version 1. =>
5 (c) 01-05-1987  By R.van der Linden "
@Setupa_e
@Setup_grid
@Setup_button
@Setup_window
Do
  Clr Special
  @Getkey
  For I=1 To 6
    Special=Special+I*Abs((X1>Sp(0,I) =>
) And (X1<Sp(1,I)) And (Y1>Sp(2,I)) A =>
nd (Y1<Sp(3,I)))
  Next I
  If Special
    Defmouse 2
    On Special Gosub Savedesktop,Read =>
desktop,Exit,Reset,Set_app
  Else
    If (X1/10>1) And (X1/10<9) And (Y =>
1/10>11) And (Y1/10<19)
      @Editgrid(Int(X1/10-1),Int(Y1/1 =>
0-11),200,50)
    Else
      If (X1>38) And (X1<601) And (Y1 =>
>199) And (Y1<326)
        @Edit_window(80,5)
      Else
        @Edita_e(X1,Y1,Key)
      Endif
    Endif
  Endif
Loop
Procedure Getkey
  Defmouse 3
  Repeat
  Until Mousek
  Repeat
    Mouse X1,Y1,Key
  Until Key=0 Or Key=3
Return
'
Procedure Exit
  Defmouse 0
  Closew 1
  Closew 0
  End
Return
'
Procedure Edita_e(X,Y,Key)
  Local I,J,Updown,O
  Clr Ud,I
  Repeat
    Inc I
    For J=0 To 1
      If (X>O(J,0,I)) And (X<O(J,1,I) =>
) And (Y>O(J,2,I)) And (Y<O(J,3,I))
        Ud=J*2-1
      Endif
    Next J
  Until Ud Or I>Index
  O=Val(Object$(1,I))
  Object$(1,I)=Str$(O-((Ud>0) And (O< =>
Abs(Val(Mx$(I,0))))+(Ud<0) And (O>0)))
  @Doset(I)
Return
'
Procedure Doset(Index)
  If Val(Mx$(Index,0))>0
    Text O(0,1,Index)+2,O(0,3,Index)- =>
3,Object$(1,Index)+"  "
  Else
    Text O(0,1,Index)+2,O(0,3,Index)- =>
3,Mx$(Index,Val(Object$(1,Index))+1)
  Endif
Return
'
Procedure Setupa_e
  Dim O(2,4,100),Object$(1,100),Mx$(1 =>
00,5)
  Restore Da
  Repeat
    Inc Index
    Object$(1,Index)="0"
    Read Object$(0,Index),M
    If M<0
      For I=1 To 1-M
        Read M$
        Mx$(Index,I)=Left$(M$+Space$( =>
8),8)
      Next I
    Endif
    Mx$(Index,0)=Str$(M)
  Until Object$(0,Index)="*"
  @Set(10,15,1,7)
  @Set(190,15,8,7)
  @Set(390,15,15,7)
  @Set(390,125,22,6)
  For I=0 To Index
    @Doset(I)
  Next I
Return
'
Procedure Set(X,Y,Index,Lght)
  Text X,Y-2,Object$(0,Index)
  For I=1 To Lght-1
    Box X,Y+(I-1)*13,X+100,Y+I*13
    Box X+15,Y+(I-1)*13,X+85,Y+I*13
    Text X+4,Y+I*13-4,Chr$(4)
    Text X+90,Y+I*13-4,Chr$(3)
    Text X+103,Y+(I*13)-4,Object$(0,I =>
ndex+I)
    O(0,0,Index+I)=X
    O(0,1,Index+I)=X+15
    O(0,2,Index+I)=Y+(I-1)*13
    O(0,3,Index+I)=Y+I*13
    O(1,0,Index+I)=X+85
    O(1,1,Index+I)=X+100
    O(1,2,Index+I)=Y+(I-1)*13
    O(1,3,Index+I)=Y+I*13
  Next I
Return
'
Da:
Data RS232 (#a),0,Duplex,-1,Full,Half =>
,Baudrate,-3,9600,4800,1200,300
Data Parity,-2,None,Odd,Even,Bits/cha =>
r,-3,8,7,6,5,Protocol,-3,None,Xon/Xoff
Data Rts/Cts,Xon+Rts,Stripbit,-1,On,O =>
ff,Printerport (#b),0,Type,-1,Dot,Daisy
Data Color,-1,B/W,Color,Pixels/line,- =>
1,1280,960,Quality,-1,Draft,Final,Port
Data -1,Printer,Modem,Papertype,-1,Fe =>
ed,Single,Control (#c),0,Mouse_response
Data 4,Keyboardclick,-1,Off,On,Bell,- =>
1,Off,On,Keyboard_pause,21
Data Keyboard_response,46,Bordercolor =>
,-1,Black,White,Menu_bar (#E),0
Data Resolution,-2,Low,Medium,High,Co =>
nfirm Copy's,-1,Off,On,Confirm Delete =>
s,-1
Data Off,On,Sort by,-3,Name,Date,Size =>
,Type,I/T Select,-1,Icon,Text,*,0
'
Procedure Editgrid(X,Y,Tx,Ty)
  Local Sc$,Sr$,E$,X1,Y1,A,V,T$
  Get 0,0,639,399,Sc$
  @Ptbox(Tx,Ty,253,233)
  Deftext ,0,0,13
  Repeat
    Text Tx+60,Ty+30,"Screen Settings"
    Text Tx+60,Ty+60,"Position ("+Str =>
$(X)+","+Str$(Y)+")"
    Text Tx+40,Ty+90,Chr$(7)+" Icon : =>
"+Gr$(Val(Grid$(0,X,Y)))
    Text Tx+40,Ty+110,Chr$(7)+" Icon_ =>
form : "+Gr$(3+Val(Grid$(1,X,Y)))
    Text Tx+40,Ty+130,Chr$(7)+" Icon_ =>
letter : "+Left$(Grid$(2,X,Y)+" ",1)
    Text Tx+40,Ty+150,Chr$(7)+" Icon_ =>
text : "+Left$(Grid$(3,X,Y)+Space$(12 =>
),12)
    Text Tx+40,Ty+170,Chr$(7)+" Exit"
    @Getkey
    A=Int((Y1-Ty-90)/20)+1
    If (X1>Tx+40) And (X1<Tx+50) And  =>
(A>=0)
      If A<2
        V=Val(Grid$(A,X,Y))
        Grid$(A,X,Y)=Str$(V-(V+1<=2+A =>
*2)+(V+1>2+A*2)*V)
      Else
        If A<4
          Get 0,0,639,399,Sr$
          @Ptbox(Tx+10,Ty+185,232,30)
          T$=Mid$("Icon_letter :",(A= =>
3)*-14)+Mid$("Icon_text    :",(A=2)*-14)
          Text Tx+15,Ty+205,T$
          Print At((Tx+130)/8,(Ty+206 =>
)/16);
          Input Grid$(A,X,Y)
          Grid$(A,X,Y)=Left$(Grid$(A, =>
X,Y),1+(A-2)*11)
          Put 0,0,Sr$
        Else
          E$="END"
        Endif
      Endif
    Endif
  Until E$="END"
  Put 0,0,Sc$
  Deftext ,0,0,6
  @Set_grid
Return
'
Procedure Set_grid
  Local X,Y,Type
  For X=0 To 7
    For Y=0 To 7
      Type=32+(Val(Grid$(0,X,Y)) And  =>
1)*36+(Val(Grid$(0,X,Y)) And 2)*26
      Text 12+X*10,118+Y*10,Chr$(Type)
    Next Y
  Next X
Return
'
Procedure Ptbox(X,Y,W,H)
  Pbox X-3,Y-3,X+W+3,Y+H+3
  Box X-2,Y-2,X+W+2,Y+H+2
  Box X,Y,X+W,Y+H
Return
'
Procedure Setup_grid
  Dim Gr$(10),Grid$(4,7,7)
  Restore Gr
  Repeat
    Read M$
    Gr$(Gindex)=Left$(M$+Space$(8),8)
    Inc Gindex
  Until M$="*"
  For I=0 To 8
    Line 10,110+I*10,90,110+I*10
    Line 10+I*10,110,10+I*10,190
  Next I
  Text 7,107,"Icon Screen"
Return
'
Gr:
Data None,Disk,Trash,Drive,Folder,Tra =>
sh,Program,Document,*
'
Procedure Savedesktop
  Local Sc$,A$,B$,C$,E$,H$,M$,I,H,X,Y
  Open "O",#1,Nam$
  Clr A$,B$
  C$=Mid$("000777000",-4*(Val(Object$ =>
(1,21))<>0),6)
  C$=C$+"7000600070055200505552220770 =>
55707505550770"
  For I=2 To 7
    A$=A$+Object$(1,I)
    B$=B$+Object$(1,I+7)
    C$=C$+Right$("00"+Object$(1,I+14) =>
,(I<7)*-(2+(I<=4)))
  Next I
  H=128*Val(Object$(1,27))+32*Val(Obj =>
ect$(1,26))+16*Val(Object$(1,25))
  E$=" "+Right$("00"+Hex$(H+8*Val(Obj =>
ect$(1,24))+3),2)+" "
  Print #1,"#a"+A$
  Print #1,"#b"+B$
  Print #1,"#c"+C$
  Print #1,"#d"+Space$(45)
  Print #1,"#E"+E$+Right$("00"+Str$(V =>
al(Object$(1,23))+1),2)
  For J=0 To 3
    I=J+Tpwdw+(J+Tpwdw>4)*4
    W$=Right$("00"+Hex$(Wi(0,I)),2)+" =>
"+Right$("00"+Hex$(Wi(1,I)),2)+" "
    W$=W$+Right$("00"+Hex$(Wi(2,I)),2 =>
)+" "+Right$("00"+Hex$(Wi(3,I)),2)+" "
    W$=W$+Right$("00"+Str$(Sgn(Wi(4,I =>
))*8),2)+" "+Wi$(I)+"@"
    Print #1,"#W 00 00 "+W$
  Next J
  Deffill ,0,0
  For X=0 To 7
    For Y=0 To 7
      If Val(Grid$(0,X,Y))
        H$="@ @"
        M$="#"+Mid$("MT",Val(Grid$(0, =>
X,Y)),1)+" "
        If (Val(Grid$(0,X,Y))=1) And =>
(Grid$(2,X,Y)<>"A") And (Grid$(2,X,Y) =>
<>"B")
          H$="@ "+Chr$(&H60)+Chr$(&H1 =>
E)+Chr$(1)+"@"
        Endif
        M$=M$+Right$("00"+Str$(X),2)+ =>
" "+Right$("00"+Str$(Y),2)+" "
        M$=M$+Right$("00"+Grid$(1,X,Y =>
),2)+" FF "+Left$(Grid$(2,X,Y)+" ",1)
        Print #1,M$+" "+Grid$(3,X,Y)+H$
      Endif
    Next Y
  Next X
  For I=0 To Apindex-1
    M$=Left$(Ap$(1,I)+Space$(2),2)+"  =>
"+Left$(Ap$(2,I)+Space$(2),2)+" "
    M$=M$+Left$(Ap$(3,I)+Space$(2),2) =>
+"  "+Ap$(4,I)+"@ "+Ap$(5,I)+"@"
    Print #1,M$
  Next I
  Print #1,Chr$(10)+Chr$(26);
  Close #1
Return
'
Procedure Readdesktop
  If Exist(Nam$)
    Clr Rindex,Windex
    Open "I",#1,Nam$
    While (Not Eof(#1)) And (Key<>26)
      Inc Rindex
```

Dat je bij de BYTE COMPUTERSHOPS terecht kunt voor de ATARI XL's, -XE's, -ST's en de daarbij behorende randapparatuur als printers, beeldschermen, modems, diskdrives en harddisks, wist je al. En dat je daarvoor zeker niet teveel betaalt bij BYTE, wist je ook. Maar dat BYTE ook de nieuwste ATARI MEGA ST met de daarbij behorende ATARI SLM laserprinter en de ATARI MS-DOS PC in zijn assortiment heeft, dat is nieuws.

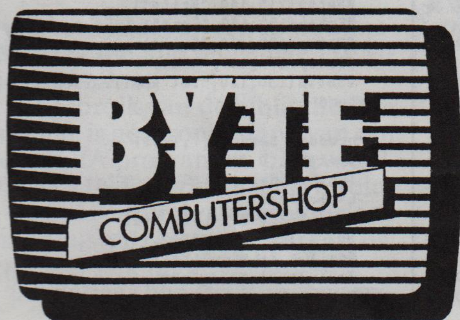
Om met de eerste te beginnen: de MEGA ST heeft maar liefst 2 Mb RAM geheugen. Zodat je ook de meest serieuze programma's zonder problemen kunt laten uitvoeren, zonder tijdverlies voor het lezen van en naar schijf. Daarbij een toetsenbord en een losse (!) systeemkast met ingebouwde floppydrive en een aparte harde schijf-unit van 20 Mb geformatteerd. Uiteraard keuze uit een mono- of RGB monitor. De onnavolgbare muis en TOS met GEM zijn standaard.

Aan deze MEGA ST kan de ATARI SLM laserprinter gekoppeld worden, die je tekst print met 8 pagina's per minuut en wel met een matrix van 300 x 300 dpi.
De derde nieuweling in ATARI-land is de ATARI MS-DOS PC. Een 8 Mhz 16 bitter met 512 Kb (of 640 Kb) aan boord èn 256 Kb video geheugen. Omschakelbare kloksnelheid, ingebouwde diskdrive 5½", HiRes monitor, muis-interface; kortom kompleet.
Maar, zeg je dan, hoe zit het met de software?

# BYTE en The new generation

Voor wat de MEGA ST betreft: volledig compatibel met de ST. Alle ST programma's draaien zonder problemen op de MEGA ST.
Maar er zijn voor de MEGA ook andere toepassingen beschikbaar. Vraag er naar als je bij BYTE komt. Op de MS-DOS PC kun je helaas niet de ST programma's draaien, maar wel alle MS-DOS software. En daarin heeft BYTE nogal wat keus! Van games tot databases en van tekstverwerking tot communicatie. Dus daar hoef je ook geen problemen mee te verwachten.
De service op de nieuwste ATARI's is weer des BYTE's: je kunt ook met de nieuwste machines in Byte's eigen ATARI-servicecentrum terecht, mocht er ooit eens iets met je machine aan de hand zijn.
En mocht je al in het bezit zijn van een computer, maar nu één van de nieuwste ATARI's willen, nou dan ruil je die ouwe toch gewoon in! BYTE geeft je er een prima inruilprijs voor.
Over prijzen gesproken: daar mogen we niet over praten. Want die zijn zoals gewoonlijk bij BYTE beschamend laag. Wil je dus weten wat die nieuwste ATARI's gaan kosten, dan moet je even bellen naar een van de BYTE Computershops. De telefoonnummers vind je onderaan. Je kunt, als je dat wilt, een Comfort Card rekening nemen, dan kun je per maand betalen.

## BYTE COMPUTERSHOP: NET IETS BETER!




**OUDE VISMARKT 29 ZWOLLE**
**TELEFOON: 038-219429**
**STEENTILSTRAAT 10 GRONINGEN**
**TELEFOON: 050-142097**
**BERLIKUMERMARKT 21 LEEUWARDEN**
**TELEFOON: 058-121210**

```
    Invoer$(Rindex)=""
    Repeat
      Key=Inp(#1)
      Invoer$(Rindex)=Invoer$(Rinde =>
x)+Chr$(Key)
    Until Key=13 Or Key=26 Or Eof(#1)
    If Left$(Invoer$(Rindex),1)=Chr =>
$(10)
      Invoer$(Rindex)=Mid$(Invoer$( =>
Rindex),2)
    Endif
  Wend
  Close #1
  @Setdesk
Else
  Alert 3,"D E S K T O P . I N F ! =>
!N O T   F O U N D ! !",1,"ABORT",Bt
Endif
Return
'
Procedure Setdesk
  Clr Windex,Apindex
  @Find("#a")
  A=Fin
  @Find("#b")
  For I=1 To 6
    Object$(1,I+1)=Mid$(Invoer$(A),I+ =>
2,1)
    Object$(1,I+8)=Mid$(Invoer$(Fin), =>
I+2,1)
  Next I
  @Find("#c")
  Object$(1,16)=Mid$(Invoer$(Fin),51,1)
  Object$(1,17)=Mid$(Invoer$(Fin),52,1)
  Object$(1,18)=Mid$(Invoer$(Fin),53,1)
  Object$(1,19)=Str$(Val(Mid$(Invoer$ =>
(Fin),54,2)))
  Object$(1,20)=Str$(Val(Mid$(Invoer$ =>
(Fin),56,2)))
  Object$(1,21)=Str$(1+(Mid$(Invoer$( =>
Fin),3,3)="000"))
  @Find("#E")
  Object$(1,23)=Str$(Val(Mid$(Invoer$ =>
(Fin),8,1))-1)
  M$=Mid$(Invoer$(Fin),4,2)
  @Conver(Mid$(Invoer$(Fin),4,2))
  Object$(1,24)=Str$(Sgn(H1 And &H8))
  Object$(1,25)=Str$(Sgn(H1 And &H10))
  Object$(1,26)=Str$(Sgn(H1 And &H20) =>
+2*Sgn(H1 And &H40))
  Object$(1,27)=Str$(Sgn(H1 And &H80))
  For I=0 To 7
    For J=0 To 7
      Grid$(0,I,J)=Str$(0)
    Next J
  Next I
  For I=1 To Rindex
    M$=Left$(Invoer$(I),2)
    If M$="#M" Or M$="#T"
      X=Val(Mid$(Invoer$(I),4,2))
      Y=Val(Mid$(Invoer$(I),7,2))
      Grid$(1,X,Y)=Str$(Val(Mid$(Invo =>
er$(I),10,2)))
      Grid$(2,X,Y)=Mid$(Invoer$(I),16,1)
      J=18
      Grid$(3,X,Y)=""
      While Mid$(Invoer$(I),J,1)<>"@"
        Grid$(3,X,Y)=Grid$(3,X,Y)+Mid =>
$(Invoer$(I),J,1)
        Inc J
      Wend
      Grid$(0,X,Y)=Str$((M$="#M")+2)
    Else
      If M$="#W"
        Inc Windex
        @Conver(Mid$(Invoer$(I),10,2))
        Wi(0,Windex)=H1
        @Conver(Mid$(Invoer$(I),13,2))
        Wi(1,Windex)=H1
        @Conver(Mid$(Invoer$(I),16,2))
        Wi(2,Windex)=H1
        @Conver(Mid$(Invoer$(I),19,2))
        Wi(3,Windex)=H1
        @Conver(Mid$(Invoer$(I),22,2))
        Wi(4,Windex)=H1
        Wi$(Windex)=Mid$(Invoer$(I),25)
        Repeat
          M=Len(Wi$(Windex))
          M$=Mid$(Wi$(Windex),M,1)
          Wi$(Windex)=Left$(Wi$(Winde =>
x),M-1)
        Until M$="@"
      Endif
      If M$="#P" Or M$="#F" Or M$="#G =>
" Or M$="#D"
        Ap$(1,Apindex)=Left$(Invoer$( =>
I),3)
        Ap$(2,Apindex)=Mid$(Invoer$(I =>
),4,3)
        Ap$(3,Apindex)=Mid$(Invoer$(I =>
),7,3)
        P=11
        Ap$(4,Apindex)=""
        Repeat
          Inc P
          Ap$(4,Apindex)=Ap$(4,Apinde =>
x)+Mid$(Invoer$(I),P,1)
        Until Mid$(Invoer$(I),P,1)="@"
        Ap$(5,Apindex)=Mid$(Invoer$(I) =>
),P+2)
        Repeat
          M=Len(Ap$(4,Apindex))
          M$=Mid$(Ap$(4,Apindex),M,1)
          Ap$(4,Apindex)=Left$(Ap$(4, =>
Apindex),M-1)
        Until M$="@"
        Repeat
          M=Len(Ap$(5,Apindex))
          M$=Mid$(Ap$(5,Apindex),M,1)
          Ap$(5,Apindex)=Left$(Ap$(5, =>
Apindex),M-1)
        Until M$="@"
        Inc Apindex
      Endif
    Endif
  Next I
  For I=0 To Index
    @Doset(I)
  Next I
  @Set_grid
  @Set_window
Return
'
Procedure Find(Find$)
  Clr Fin
  Repeat
    Inc Fin
  Until (Fin=Rindex) Or Left$(Invoer$ =>
(Fin),2)=Find$
Return
'
Procedure Conver(Hx$)
  Local K,H
  Clr H1
  For K=1 To Len(Hx$)
    H$=Mid$(Hx$,K,1)
    H1=16*H1+Asc(H$)+(H$<"A")*Asc("0" =>
)+(H$>="A")*(Asc("A")-10)
  Next K
Return
'
Procedure Setup_window
  Wdwnr=1
  Tpwdw=1
  Dim Wi$(4),Wi(4,4)
  Text 265,196,"Window Screen"
  @Set_grid
  For I=0 To 80
    Line 39+I*7,200,39+I*7,200+25*5
    Line 39,200-(I<26)*I*5,39+7*80,20 =>
0-(I<26)*I*5
  Next I
  Get 0,230,639,399,Wndw$
Return
'
Procedure Set_window
  Put 0,230,Wndw$
  For J=1 To 4
    I=J+Tpwdw+(J+Tpwdw>4)*4
    Deffill 1,2,I
    If Wi(4,I)
      Pbox 39+7*Wi(0,I),200+5*Wi(1,I) =>
,39+7*(Wi(0,I)+Wi(2,I)),200+5*(Wi(1,I =>
)+Wi(3,I))
    Endif
  Next J
  Deffill ,0,0
Return
'
Procedure Edit_window(X,Y)
  Local Sc$
  Clr E$
  Get 0,0,639,399,Sc$
  @Ptbox(X,Y,500,180)
  Deftext ,0,0,13
  Text X+200,Y+20,"Window Settings"
  Repeat
    Text X+30,Y+50," "+Chr$(5)+"  Win =>
dow 1 On/Off : "+Mx$(7,2-Sgn(Wi(4,1)))
    Text X+30,Y+70," "+Chr$(5)+"  Win =>
dow 2 On/Off : "+Mx$(7,2-Sgn(Wi(4,2)))
    Text X+30,Y+90," "+Chr$(5)+"  Win =>
dow 3 On/Off : "+Mx$(7,2-Sgn(Wi(4,3)))
    Text X+30,Y+110," "+Chr$(5)+"  Wi =>
ndow 4 On/Off : "+Mx$(7,2-Sgn(Wi(4,4)))
    Text X+30,Y+130," "+Chr$(5)+"  Ed =>
it Window      : "+Str$(Wdwnr)
    Text X+30,Y+150," "+Chr$(5)+"  To =>
p Window       : "+Str$(Tpwdw)
    Text X+30,Y+170," "+Chr$(5)+"  Wi =>
ndow Path    : "+Left$(Wi$(Wdwnr)+Sp =>
ace$(30),30)
    Text X+270,Y+50,Chr$(4)+" "+Chr$( =>
3)+" X-position      : "+Str$(Wi(0,Wd =>
wnr))+"  "
    Text X+270,Y+70,Chr$(4)+" "+Chr$( =>
3)+" Y-Position      : "+Str$(Wi(1,Wd =>
wnr))+"  "
    Text X+270,Y+90,Chr$(4)+" "+Chr$( =>
3)+" Width           : "+Str$(Wi(2,Wd =>
wnr))+"  "
    Text X+270,Y+110,Chr$(4)+" "+Chr$ =>
(3)+" Height      : "+Str$(Wi(3,W =>
dwnr))+"  "
    Text X+270,Y+150," "+Chr$(5)+"   E =>
xit"
    Repeat
      @Getkey
      Yp=Int((Y1-40)/20)+1
      Xp=((X1>115) And (X1<125))*-1+( =>
(X1>350) And (X1<370))*-2
      Ud=(X1>345) And (X1<360)+((X1>3 =>
59) And (X1<380))*-1
      If (Yp>0) And (Yp<8) And Xp
        If Xp=1
          Fl=1
          On Yp Gosub Wi,Wi,Wi,Wi,Ew, =>
Tw,Wp
        Else
          Fl=0
          If Yp=6
            E$="END"
          Else
            On Yp Gosub Xps,Yps,Wps,Hps
          Endif
        Endif
      Endif
    Until E$="END" Or Fl
  Until E$="END"
  Deftext ,0,0,6
  Put 0,0,Sc$
  @Set_window
Return
'
Procedure Xps
  @Dopos(0,80)
Return
'
Procedure Yps
  @Dopos(1,25)
Return
'
Procedure Wps
  @Dopos(2,80)
Return
'
Procedure Hps
  @Dopos(3,25)
Return
'
Procedure Dopos(Wnr,Max)
  If ((Ud>0) And (Wi(Wnr,Wdwnr)<Max)) =>
 Or ((Ud<0) And (Wi(Wnr,Wdwnr)>0))
    Wi(Wnr,Wdwnr)=Wi(Wnr,Wdwnr)+Ud
  Endif
  @Set_window
  Text X+270+22*8,Y+50+20*Wnr,Str$(Wi =>
(Wnr,Wdwnr))+"  "
Return
'
Procedure Wi
  Wi(4,Yp)=Abs(Sgn(Wi(4,Yp))-1)
  @Set_window
Return
'
Procedure Ew
  Wdwnr=Wdwnr+Wdwnr*(Wdwnr>3)+1
Return
'
Procedure Tw
  Tpwdw=Tpwdw+Tpwdw*(Tpwdw>3)+1
  @Set_window
Return
'
Procedure Wp
  Tx=80
  Ty=219
  Get 0,0,639,399,Sr$
  @Ptbox(Tx,Ty,500,30)
  Text Tx+15,Ty+20,"Window Path :"
  Print At((Tx+130)/8,(Ty+21)/16);
  Input Wi$(Wdwnr)
  Put 0,0,Sr$
Return
'
Procedure Setup_button
  @Setsp(215,140,"Write")
  @Setsp(150,140,"Read")
  @Setsp(150,170,"Exit")
  @Setsp(290,140,"Reset")
  @Setsp(150,110,"Set Application")
Return
'
Procedure Setsp(X,Y,Txt$)
  Deftext ,0,0,13
  Inc Spindex
  L=Len(Txt$)
  Pbox X-3,Y-3,X+L*10,Y+20
  Box X-1,Y-1,X+L*10-2,Y+18
  Box X,Y,X+L*10-3,Y+17
  Text X+L-1,Y+14,Txt$
  Sp(0,Spindex)=X-3
  Sp(1,Spindex)=X+L*10
  Sp(2,Spindex)=Y-3
  Sp(3,Spindex)=Y+20
```

```
  Deftext ,0,0,6
Return
'
Procedure Reset
  Clr Rindex,Windex
  Restore Rs
  Repeat
    Inc Rindex
    Read Invoer$(Rindex)
  Until Invoer$(Rindex)="*"
  Reset=1
  @Setdesk
Return
'
Rs:
Data #a020000,#b001000
Data #c7770007000600070055200505555222 =>
07705570750555077031111 0301
Data #d,#E 7B 03,#W 00 00 02 04 4C 13 =>
 08 A:\*.*@
Data #W 00 00 0D 08 2A 0B 00 @,#W 00  =>
00 0E 09 2A 0B 00 @
Data #W 00 00 0F 0A 2A 0B 00 @,#M 00  =>
00 00 FF A FLOPPY DISK@ @
Data #M 01 00 00 FF B FLOPPY DISK@ @, =>
#M 02 00 00 FF D RAM DISK@ @
Data #T 07 00 02 FF   TRASH@ @,#F FF  =>
04  @ *.*@,#D FF 01   @ *.*@
Data #G 03 FF   *.APP@ @,#G 03 FF   * =>
.PRG@ @,#F 03 FF   *.TOS@ @
Data #P 03 FF   *.TTP@ @,*
'
Procedure Set_app
  Local Sc$
  Clr E$
  Get 0,0,639,399,Sc$
  Clearw 1
  Deftext ,0,0,13
  Defmouse 3
  X1=100
  Y1=100
  Pbox X1-3,Y1-53,642-X1,402-Y1
  Box X1,Y1-50,639-X1,399-Y1
  Box X1-1,Y1-51,640-X1,400-Y1
  Line 619-X1,Y1-50,619-X1,399-Y1
  Line X1+50,Y1-50,X1+50,379-Y1
  Line X1+99,Y1-50,X1+99,379-Y1
  Line X1+151,Y1-50,X1+151,379-Y1
  Line X1+285,Y1-50,X1+285,379-Y1
  Line X1,Y1-31,639-X1,Y1-31
  Line X1,Y1-30,619-X1,Y1-30
  Line X1,379-Y1,639-X1,379-Y1
  Line 619-X1,Y1-11,639-X1,Y1-11
  Line 619-X1,359-Y1,639-X1,359-Y1
  For I=Y1-30 To 379-Y1 Step 16
    Line X1,I,619-X1,I
  Next I
  Text 626-X1,Y1-35,Chr$(1)
  Text 626-X1,394-Y1,Chr$(2)
  Text 622-X1,Y1-16,Chr$(1)+Chr$(1)
  Text 622-X1,375-Y1,Chr$(2)+Chr$(2)
  Text X1+14,Y1-35," # "
  Text X1+51,Y1-35,"Icon 1"
  Text X1+102,Y1-35,"Icon 2"
  Text X1+195,Y1-35,"Path 1"
  Text X1+330,Y1-35,"Path 2"
  Text 300,295,"Exit"
  @Install(619-X1,639-X1,Y1-50,Y1-31,1)
  @Install(619-X1,639-X1,379-Y1,399-Y =>
1,4)
  @Install(X1,X1+50,Y1-31,379-Y1,5)
  @Install(X1+51,X1+99,Y1-31,379-Y1,6)
  @Install(X1+100,X1+151,Y1-31,379-Y1,7)
  @Install(X1+152,X1+285,Y1-31,379-Y1,8)
  @Install(X1+286,619-X1,Y1-31,379-Y1,9)
  @Install(619-X1,639-X1,Y1-31,Y1-11,2)
  @Install(619-X1,639-X1,359-Y1,379-Y =>
1,3)
  @Install(100,515,280,300,10)
  Pstion=0
  @Show(Pstion)
  Repeat
    Repeat
    Until Mousek
    Repeat
      Mouse X,Y,Key
    Until Key=0 Or Key=3
    Yp=Int((Y-70)/16)
    F=0
    For I=1 To 10
      If (X>Muis(I,1)) And (X<Muis(I, =>
3)) And (Y>Muis(I,2)) And (Y<Muis(I,4))
        F=I
      Endif
    Next I
    On F Gosub U1,U2,D2,D1,S1,S2,S3,S =>
4,S5,End
  Until E$="END"
  Put 0,0,Sc$
Return
'
Procedure Install(X1,X2,Y1,Y2,Index)
  Muis(Index,1)=X1
  Muis(Index,2)=Y1
  Muis(Index,3)=X2
  Muis(Index,4)=Y2
Return
'
Procedure End
  E$="END"
  Deftext ,0,0,6
Return
'
Procedure Show(P)
  Deftext ,8,0,13
  For I=0 To 12
    If I+P<Apindex
      Text X1+17,84+I*16,Left$(Ap$(1, =>
I+P)+Space$(2),2)
      Text X1+65,84+I*16,Left$(Ap$(2, =>
I+P)+Space$(2),2)
      Text X1+115,84+I*16,Left$(Ap$(3 =>
,I+P)+Space$(2),2)
      Text X1+154,84+I*16,Left$(Ap$(4 =>
,I+P)+Space$(15),15)
      Text X1+288,84+I*16,Left$(Ap$(5 =>
,I+P)+Space$(15),15)
    Else
      If I+P<=Apindex
        Text X1+17,84+I*16,Space$(2)
        Text X1+65,84+I*16,Space$(2)
        Text X1+115,84+I*16,Space$(2)
        Text X1+154,84+I*16,Space$(15)
        Text X1+288,84+I*16,Space$(15)
      Endif
    Endif
  Next I
Return
'
Procedure U1
  Pstion=Pstion+(Pstion>0)
  @Show(Pstion)
Return
'
Procedure U2
  Pstion=Pstion+13*(Pstion>13)
  @Show(Pstion)
Return
'
Procedure D1
  Pstion=Pstion-(Pstion<Apindex)
  @Show(Pstion)
Return
'
Procedure D2
  Pstion=Pstion-13*(Pstion<Apindex-13)
  @Show(Pstion)
Return
'
Procedure S1
  @Input(1,Yp+Pstion,2)
Return
'
Procedure S2
  @Input(2,Yp+Pstion,2)
Return
'
Procedure S3
  @Input(3,Yp+Pstion,2)
Return
'
Procedure S4
  @Input(4,Yp+Pstion,20)
Return
'
Procedure S5
  @Input(5,Yp+Pstion,20)
Return
'
Procedure Input(Xindex,Yindex,Maxlen)
  Deftext ,0,0,13
  If Yindex<=Apindex
    If Yindex=Apindex
      Inc Apindex
    Endif
    Tx=20
    Ty=315
    Get 0,0,639,399,Sr$
    @Ptbox(Tx,Ty,600,30)
    T$="Old :"+Ap$(Xindex,Yindex)+"     =>
  New : "
    Text Tx+15,Ty+20,T$
    Print At(Tx/8+3+Len(T$),(Ty+21)/16);
    Input Ap$(Xindex,Yindex)
    Put 0,0,Sr$
    @Show(Pstion)
  Endif
Return
```

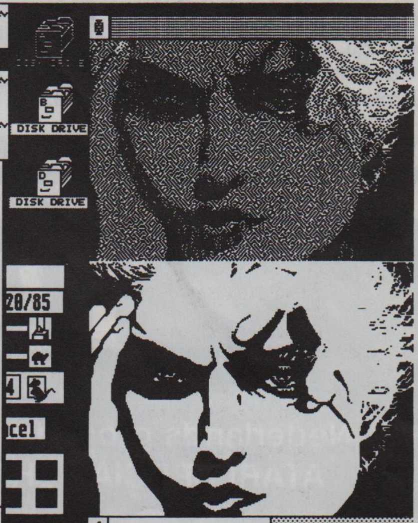
Vensters, menubalken met illustraties, dialoogschermen en ikonen waren tot nu toe niet weggelegd voor de GfA programmeur. De instruktieset van GfA Basic biedt immers onvoldoende kommando's voor echte GEM-programma's.
Egon Lubbers brengt hier verandering in. Hij verzamelde en programmeerde START's nieuwe specialdisk: TOOLKIT GEM.

Het pakket bestaat uit een subroutinebibliotheek die alle mogelijke GEM-gebieden beslaat.
Om het gebruik van de bibliotheek te illustreren, is het bronbestand van een volledig GfA programma bijgevoegd. MONOPRINT werkt in alle resoluties, maar levert in hoge resolutie een extra bonus op: zo mooi werden uw tekeningen nog nooit afgedrukt!
De uitvoerige handleiding becommentarieert het voorbeeld, zodat u in een ommezien echt in GEM leert programmeren.

Abonnees betalen f 49,- voor TOOLKIT GEM. Niet abonnees wordt f 69,- in rekening gebracht.




Door
Egon Lubbers



# OFFICIAL ATARI ST-DEALER

## ZAKELIJKE SOFTWARE

**• DATABASES**

| | |
|---|---|
| Astodat II | f 299,- - |
| ST Datamanager | f 299,- - |
| DB Man | f 359,- - |
| Superbase Personal | f 449,- - |
| Datamat | f 127,50 |
| K-data | f 198,- - |

**• FINANCIAL SOFTWARE**

| | |
|---|---|
| Projecta-4 ST | f 829,- - |
| Projecta-4 ST + Factura ex BTW | f 895,- - |
| Marko Boek | f 369,- - |

**• DIVERSEN**

| | |
|---|---|
| K-Switch | f 109,- - |
| K-Graph II | f 198,- - |
| VIP 1.2.3 (GEM) | f 798,- - |
| Swift Calc | f 299,- - |
| Logistix | f 495,- - |
| Micro Time (int.cl.card) | f 199,- - |

**• TEKSTVERWERKERS**

| | |
|---|---|
| Signum | f 549,- - |
| 1st Word Plus | f 239,- - |
| K-word | f 179,- - |
| Word Writer | f 299,- - |

**• MODEM SOFTWARE**

| | |
|---|---|
| Viewdata ST | f 298,- - |
| PC Intercom | f 249,- - |
| K-Comm V 2.0 | f 179,- - |
| Mi-Term | f 89,- - |

**• PROGRAMMEERTALEN**

| | |
|---|---|
| G.F.A. Basic 2.0 | f 215,- - |
| G.F.A. Basic Compiler | f 210,- - |
| GST Macro Assembler | f 149,- - |
| Devpac Hisoft | f 189,- - |
| Metacomco Pascal | f 349,- - |
| U.C.S.D. Pascal | f 498,- - |
| Mark Williams C | f 599,- - |
| Lattice C Comp. | |
| Metacomco | f 398,- - |
| Modula 2 | f 449,- - |
| Pro-Fortran | f 549,- - |
| Fast Basic | f 399,- - |
| L.D.W. Basic | f 199,- - |
| Macro Assembler | f 199,- - |
| St Pascal | f 298,- - |
| Megamax C | f 699,- - |
| B.C.P.L. | f 399,- - |
| Cambridge Lisp | f 549,- - |
| Prolog | f 249,- - |

**• PROGRAMMEER UTILITIES**

| | |
|---|---|
| K-Resource (Res. Constr. Set) | f 149,- - |
| Modula 2 Toolkit | f 199,- - |

**• UTILITIES ALGEMEEN**

| | |
|---|---|
| Super Directory | f 99,- - |
| R.A.I.D. | f 99,- - |
| ST Multi Hardc. | f 155,- - |
| Flash Back | f 198,- - |
| The Alternative | f 49,- - |

**• DESK ACC**

| | |
|---|---|
| Cornerman | f 79,- - |
| Macro Manager | f 149,- - |
| Desk Assist | f 199,- - |

**• TEKENPROGRAMMA'S**

| | |
|---|---|
| Stad | f 239,- - |
| Monostar Plus | f 179,- - |
| Degas Elite | f 189,- - |
| Art Director | f 179,- - |
| Cad-3D | f 149,- - |
| G.F.A. Draft Plus | f 449,- - |
| G.F.A. Vector | f 189,- - |
| Easy Draw | f 298,- - |
| Type Setter Elite | f 119,- - |
| Fleetstreet Publisher | f 459,- - |
| Sprite Construction Kit | f 79,- - |
| Printmaster | f 129,- - |
| Printmaster Lib = 1 | f 99,- - |
| Color Star | f 129,- - |
| Film Director | f 199,- - |
| Future Design Dsk | 79,- - |
| G.F.A. Object | f 349,- - |
| Aegis Animator | f 199,- - |
| Text Design ST | f 127,50 |
| Publishing Partner | f 499,- - |

**• DIGITIZERS**

| | |
|---|---|
| Turbo Dizer | f 798,- - |
| Arttizer | f 498,- - |
| Print & Technik | f 849,- - |

**• MUZIEKPROGRAMMA'S**

| | |
|---|---|
| Music Studio | f 139,- - |
| G.I.S.T. (Sound Tool) | f 199,- - |
| ST Replay | f 329,- - |
| Midiplay | f 99,- - |
| Digidrum | f 159,- - |

**• ADVENTURES**

Infocom adventures uit voorraad leverbaar

| | |
|---|---|
| Hacker II | f 119,- - |
| Mind Shadow | f 119,- - |
| Guild of Thieves | f 99,- - |
| Phantasie II | f 129,- - |

| | |
|---|---|
| Exodus Ultima II | f 99,- - |
| Borrowed Time | f 119,- - |
| Passengers of the wind | f 109,- - |

**• GAMES SPORT**

| | |
|---|---|
| World Games | f 79,95 |
| Bridge Player 2000 | f 79,50 |
| Tech Mate Chess | f 79,- - |
| Leader Board | f 99,- - |
| Psion Chess | f 99,- - |

**• GAMES ALGEMEEN**

| | |
|---|---|
| Star Glider | f 99,- - |
| Goldrunner | f 99,- - |
| Balance of Power | f 139,- - |
| Deep Space | f 129,- - |
| Metro Cross | f 99,- - |
| Gato | f 109,- - |
| Flight Simulator II | f 179,- - |
| Macadam Bumper | f 99,- - |
| Prohibition | f 99,- - |
| Sub Battle Simulator | f 99,- - |
| Airball | f 99,- - |
| Barbarian | f 99,- - |
| Jupiter Probe | f 59,- - |
| Silent Service | f 99,- - |
| Crafton | f 99,- - |
| S.D.I. | f 99,- - |
| Extravaganza | f 49,95 |
| Trail Blazer | f 99,- - |
| Gauntlet | f 99,- - |
| Arkanoid | f 69,- - |
| Road Runner | f 99,- - |

**• GAMES EDUCATIEF**

| | |
|---|---|
| Rekenen voor kl 1 | f 69,- - |
| Rekenen voor kl 2 t/m 6 | f 69,- - |

## HARDWARE

**• HARDWARE**

| | |
|---|---|
| Atari 1040 STf/SM125 | f 1998,- - |
| Atari 1040 STf/SC1224 | f 2598,- - |
| Atari 520 STm/SF354 | f 999,- - |
| Modulator ST | f 269,- - |
| Mouse matjes v.a. | f 12,95 |
| Compact kast | f 169,- - |

*Leveringen geschieden onder rembours door geheel Nederland. De rembourskosten variëren van f 12,50 voor programmatuur tot f 25,00 voor hardware. Eventuele prijswijzigingen en zetfouten voorbehouden.*

**OFFICIAL ATARI REPAIR CENTER**

**Prijswijzigingen voorbehouden. Alle prijzen incl. BTW**

# TELEKODER

**COMPUTERS**

**Nederlands grootste computer speciaalzaak-Verzending door heel Nederland**

**ATARI SPECIAL SHOP TEL. 010-4111325* Hoogstraat 26-53-65 Rotterdam**

***Kwam kwaliteitssoftware voor de ST tot dusver uit Duitsland en in mindere mate uit Engeland en Amerika, Nederland spreekt de laatste tijd ook een woordje mee.***
***Twee Amsterdammers, Gert Poletiek en de voor trouwe START-lezers niet onbekende Jos Vermaseren, hebben programma's ontwikkeld die elk rustig de naam geavanceerd mogen dragen.***
***Een van de belangrijkste betatesters van de produkten, Daan Josephus Jitta, leidt deze professionele software bij u in.***

# GP-SHELL
# de kracht van toetsenbordkommando's

## SHELL HELPT

De GP-Shell is een programma dat een Unix-achtige omgeving produceert op de ST. De letters 'GP' zijn simpelweg de initialen van auteur Gert Poletiek.
Wie een shell gebruikt, heeft de desktop in feite niet meer nodig (behalve dan om het shell-programma te starten). De shell kan allerlei kommando's krijgen: een file kopiëren, een programma starten, etc.
De desktop is een uitstekende omgeving voor beginnende gebruikers of mensen die de ST incidenteel, of slechts voor één programma nodig hebben. Maar iedereen die de ST wat vaker te hulp roept, zal van tijd tot tijd de gelikte GEM-omgeving wel eens verwensen en willen dat het ook anders kan.
Wat te denken van een enkele toetsaanslag om het laatste kommando nogmaals uit te voeren (kompleet met alle op te geven 'TTP' parameters)? Of om in één klap een directory, vijf niveaus diep, op een andere disk in te gaan? Voor dit laatste moet je in de desktop vijf keer dubbel op een ikoon klikken en wachten!
Voor mensen met een harddisk is het mogelijk om allerlei programma's in een groot aantal verschillende directories te bewaren en, zonder precies te weten waar een specifiek programma staat, het toch te starten.

## WASLIJST

De GP-shell biedt een groot aantal ingebouwde kommando's:
alias, alloc, bell, cat, clear, chmod, cp, cursor, dirs, df, echo, errorfile, find, font, free, hashtable, help, history, keyclick, keydelay, keyrepeat, logout, lpr, ls, man, mkdir, mount, palette, popd, printenv, pushd, pwd, ramdisk, read, rehash, rm, rmdir, set, setenv, sleep, touch, umount, unalias, unset, unsetenv, verify, which, who.
Een hele waslijst! Zoals zelfs een met Unix onbekende lezer kan zien, zijn er kommando's om instellingen te regelen (bell, cursor, keyclick/delay/repeat, palette).
Daarnaast bevat de lijst kommando's om file-operaties af te handelen:

**cp** - kopieert een file of een directory
**df** - geeft informatie over de bezetting van een disk (zonder crashes bij een groot aantal folders)
**ls** - laat de inhoud van een directory volgens allerlei kriteria zien (verborgen files, archives, etc.)
**mkdir** - maakt een directory
**rm** - verwijdert een file of een komplete directory
**rmdir** - verwijdert een (lege) directory

## HULPPROGRAMMA'S

Ook zijn er externe kommando's. Externe kommando's zijn kommando's die niet in de shell geïntegreerd zijn, maar als losse programmaatjes worden meegeleverd.
Uiteraard kan iedereen zijn eigen externe kommando's toevoegen, maar de volgende worden bijgeleverd:
cmp, du, dump, envtest, fastdisk, format, make, more, normdisk, od, size, wc
Een paar speciale kommando's:
**more** - toont een file op het scherm. Dit is natuurlijk ook via de desktop mogelijk, maar more kan veel meer (zoals kursief en dik weergeven) en is bovendien supersnel.
**verify** - schakelt met één kommando de verify-flag voor schrijfoperaties aan of uit.
**pushd** - ga naar een andere directory, maar bewaar de huidige in een popd buffer. Later kan de gebruiker met één kommando weer terug naar de oude directory.
**fastdisk** - verandert een aantal parameters waardoor disk-operaties sneller verlopen (maar ook iets minder betrouwbaar). Blijkt in de praktijk veilig genoeg: ik heb geen problemen ondervonden.




**format** - formatteert disks enkel- of dubbelzijdig met negen of tien sektoren per track. Tien sektoren per track geeft (bij dubbelzijdige schijven) een opslagruimte van 791 kByte.
**ramdisk** - de GP-shell heeft een kommando om een tijdelijke ramdisk aan te maken. Deze ramdisk kan later weer met één kommando worden weggegooid en is bijzonder geschikt voor allerlei hulp-files. Een nadeel is dat de ramdisk bij het verlaten van de shell verdwijnt. Dit was noodzakelijk in verband met geheugenbeheer, maar het blijft jammer.

## SHELL SCRIPT

Alle soorten programma's worden ondersteund. Bestanden met de extensie APP, GEM, TTP en TOS kunnen gestart worden door de naam van het programma in te tikken. De extensie hoeft niet te worden opgegeven, maar het mag wel.
Er is nog een vijfde 'type' file voor de GP-shell: een SH-extensie. Een dergelijke file wordt een shell script genoemd. In zo'n file kunnen zowel interne als externe kommando's voorkomen, die worden uitgevoerd als de naam van de file wordt ingetypt. Een shell script is dus vergelijkbaar met een batch file. De shell verricht hier echter al het werk: een programma als BATCH.TTP is dus niet nodig.
Helaas beschikt de shell niet over kommando's als if-then-else en goto; dit is een minpunt ten opzichte van een aantal konkurrenten.
Programma's (of shell scripts) hoeven niet in de directory te staan waarin de gebruiker zich op dat moment bevindt. Het is namelijk mogelijk om een rijtje directories op te geven. Die directories worden achtereenvolgens afgezocht tot het opgegeven programma is gevonden. Dit zoeken gaat razendsnel, omdat bij het opstarten van de shell een hashtable wordt gecreëerd. De variabele die de shell voor dit werk gebruikt heet 'path'. Er is ook een 'cdpath' variabele, die in een gedefinieerd rijtje directories zelf een subdirectory zoekt, zodat de gebruiker zich ook niet meer hoeft te bekommeren waar een directory precies staat.

## UNIX-SAUS

De auteur heeft er duidelijk naar gestreefd om de shell met een stevige Unix-saus te overgieten. Dat begint al gelijk. De shell meldt zich keurig met:
Atari ST / GP-shell
login:
Daarop kan een persoonlijke key worden ingetikt, waarna om een password wordt gevraagd. Dit password bewaart de shell overigens gewoon in een file, dus veilig is het systeem niet.
Directories in een path-naam kunnen naar keuze met de voor ST-gebruikers bekende backslash (\) of met een slash (/) gescheiden worden. Hier heeft geen enkel programma last van: de GP-shell konverteert steeds de / naar een \. De eerste slash na een disk identifier hoeft zelfs niet ingetikt te worden. Een path kan dus zijn: A:START/ARTIKEL.
In tegenstelling tot een groot aantal konkurrenten beschikt de GP-shell over een 'environment'. Als de gebruiker environment variabelen zet, dan worden deze meegegeven aan het te starten programma. Andersom worden alle argumenten die een programma meekrijgt, op dezelfde manier ook in het environment opgenomen. Het voordeel hiervan is dat de limiet van 127 karakters voor de 'command tail' komt te vervallen: de limiet wordt bepaald door de beschikbare hoeveelheid geheugen (deze truuk is afgekeken van de Mark Williams C compiler).
Bij de shell wordt een make geleverd die volledig geïntegreerd is. Dit wil zeggen dat alle bestaande shell-kommando's in de make-files kunnen worden opgenomen. Ook make maakt hierbij gebruik van de ingestelde variable 'path'.

## AMERIKA

De GP-shell heeft een ingebouwde fontloader en wordt standaard geleverd met een zestal zelfgemaakte fonts in verschillende formaten (8x8, 8x12, 8x16). De IBM- en Macintosh-achtige fonts zijn bijzonder geslaagd. Het font-formaat is standaard VDI.
De shell bevat ook een 'killer': met een druk op de control, alternate en rechter shift-toets worden een heleboel programma's 'grof' onderbroken. Grof staat tussen apostrofs omdat, anders dan bij sommige andere killers, wel gekeken wordt wat het te killen programma doet. Als de software het operating systeem aanspreekt, doet de killer niets. Dit voorkomt hangups. Vaak

kan een (TOS) programma overigens ook al met control-C worden afgebroken.
De 'command editing' is uitstekend verzorgd. Met de pijltoetsen kunnen eerder gegeven kommando's teruggehaald, veranderd en opnieuw uitgevoerd worden.
De GP-shell is verder volledig kompatibel met alle richtlijnen die Atari bekend heeft gemaakt. Dit is mede het gevolg van het feit dat een aantal mensen binnen Atari in de Verenigde Staten (onder andere Alan Pratt) de GP-shell al gedurende maanden gebruikt en geregeld advies heeft gegeven.
De levering is kompleet: naast het programma, de hulp-programma's en de fonts bevindt zich ook een C-source op de disk. Dit is de listing van een programma dat de shell aanroept om kommando's uit te voeren. Zo is het mogelijk vanuit een zelf geschreven programma de onderliggende shell aan te roepen. STedi, het pakket dat hierna wordt besproken, maakt hier gebruik van.
De handleiding van de GP-shell heeft een typische Unix-layout: een opsomming van alle beschikbare interne en externe kommando's in alfabetische volgorde. Er wordt nooit meer dan één kommando per bladzijde behandeld. Een kommando opzoeken gaat daardoor snel. En mocht er ondanks het goede handboek toch behoefte aan meer informatie bestaan: Unix-boeken zijn er genoeg!

## ☆KONKLUSIE

*Het is erg jammer dat het niet mogelijk is if-then-else konstrukties in een shell script op te zetten. Hopelijk zal dit in een volgende versie toegevoegd worden. Ook verdwijnt heel af en toe de muis als een programma opstart. Vermoedelijk wordt dit probleem veroorzaakt door fouten in GEMDOS.*
*De GP-shell is een erg betrouwbaar produkt. De afwerking is perfekt. Ik gebruik deze shell al lange tijd en heb met geen enkel programma problemen gehad. Het is prettig werken in de door de GP-shell geschapen omgeving. Daar ik andere Unix-achtigen (Megamax, Mark Williams, Beckmeyer) in het verleden wel eens in handen heb gehad, durf ik te beweren dat de GP-shell qua prestaties ruim boven zijn direkte konkurrenten uitsteekt. Prijsverschil is er vrijwel niet: de shell kost f 195,-.*

# STedi turbo-editor in assembler

## PUUR ASCII

Een editor is een soort tekstverwerker die hoofdzakelijk gebruikt wordt om programma's in te tikken. Een tekst rechts uitlijnen kan een editor meestal niet, omdat dit voor programma's een overbodige optie is. Een editor kan natuurlijk ook voor andere zaken nuttig zijn: ik gebruik zelf altijd een editor om teksten in te tikken voor TeX, een zeer geavanceerde tekstverwerker die ASCII-bestanden als input nodig heeft.
Veel mensen zullen op dit moment met behulp van First Word programma's intikken, waarbij ze dan de WP (=Word-Processor) mode afzetten. First Word schrijft de file dan weg in pure ASCII-vorm (alleen de letters en cijfers). Dit in tegenstelling tot een file die in WP-mode wordt weggeschreven: die bevat allerlei extra informatie voor de tekstverwerker, zoals einde van een paragraaf, het gebruikte lettertype (dik, kursief), lengte en breedte van een pagina, etc. Het zal duidelijk zijn dat dit soort informatie voor programma's niet gewenst is.
Het zwaartepunt bij het ontwerpen van een tekstverwerker ligt meestal bij het tekstverwerken zelf. De ASCII-stand is een extraatje. Vaak is het uiterlijk van een tekstverwerker ("WYSIWIG", What You See Is What You Get) belangrijker dan de snelheid.
Een editor hoeft daarentegen alleen maar karakters op het scherm te laten zien en is meestal meer op snelheid geschreven. De programmeur moet van voor naar achter door de file heen kunnen springen. Daarnaast heeft een editor vaak hele handige kommando's voor het bewerken van de listings.
Voor iemand die heel af en toe een programma schrijft, is het waarschijnlijk handiger om zijn of haar favoriete tekstverwerker te gebruiken, zodat er geen nieuwe kommando's geleerd hoeven te worden. Maar iemand die wat regelmatiger programma's schrijft, zal de snelheid en flexibiliteit van een editor belangrijk genoeg vinden om hem te gebruiken.

## 100% ASSEMBLER

Er zijn natuurlijk een heleboel editors op de markt. Wat maakt STedi nu tot zo'n bijzondere? In de eerste plaats de enorme snelheid waarmee de kommando's worden uitgevoerd. Dit komt voornamelijk doordat STedi geheel in assembler is geschreven en de programmeur altijd goed zijn koppie heeft gebruikt bij snelheidskritische onderdelen. Verder beschikt STedi over een groot aantal kommando's, waarvan enkele uniek zijn. Tot slot mag het feit dat STedi als desk-accessory en als los programma wordt geleverd wel worden opgemerkt: op deze manier kan de gebruiker zelf bepalen welke vorm voor hem of haar het handigst is. Met name mensen die altijd de desktop gebruiken, zullen het desk-accessory waarschijnlijk zeer waarderen!

## HONDERDDUIZEND REGELS

Als STedi wordt gestart, meldt de fileselector zich als eerste. De gebruiker kan een file naar wens kiezen of de fileselector gewoon passeren door meteen op return te drukken. Daarna wordt het scherm van STedi zichtbaar. Om de snelheid van STedi bij het scrollen van het scherm zo hoog mogelijk te houden, is besloten niet van de diensten van het (vrij trage) GEM gebruik te maken. Toch is het mogelijk met de muis te werken. Hierover later meer.
De eerste 24 regels van het scherm zijn bestemd voor de file die ingetypt of bewerkt wordt. De laatste regel op het scherm (de statusregel of statusbalk) heeft een kontrasterende kleur.
Helemaal aan de linkerkant van de statusbalk bevinden zich vijf posities waarin het regelnummer aangegeven wordt. Een file mag, als er voldoende geheugen beschikbaar is, maximaal 99999 regels lang zijn. Naar ik mag aannemen voor iedereen voldoende! Daarna, meteen achter het copyright-teken, volgt een ruimte om kommando's in te tikken.
Aan de rechterkant van de statusbalk bevindt zich een aantal status-indicatoren: deze geven aan of STedi in insert- of overwrite-mode staat, in Atari-, Unix- of hexadecimale mode, de richting en het bereik van de search- en replace opdrachten, of deze laatste opdrachten hoofdlettergevoelig zijn of niet en welke buffer er gebruikt wordt voor kopieeropdrachten. Ook een indicator voor de Caps-lock toets ontbreekt niet. Uiterst rechts staat de naam van de file die zich op dat moment op het scherm bevindt.
Kommando's kunnen op verschillende manieren worden gegeven: met de "Control"-toets, met de "Alternate"-toets, via de "Escape"-toets ingetikt op de statusregel en met behulp van de funktietoetsen. Ergens is het wel jammer dat het programma alle funktietoetsen in beslag neemt. Ze zijn altijd erg geschikt om door de gebruiker zelf geprogrammeerd te worden. Deze heeft daarvoor nu zogenaamde learn-buffers (control-0 tot en met control-9).
STedi bevat tien tekst-buffers: de buffers één tot en met acht kunnen elk een file bevatten. De laatste twee buffers (negen en nul) zijn de zogenaamde yank-buffers: hier wordt een stuk tekst bewaard dat uit een van de andere buffers is gehaald om ergens anders weer toegevoegd te kunnen worden (cut/copy/paste-kommando's). In deze yank-buffers kunnen overigens alle kommando's van STedi gewoon gebruikt worden.
Na een druk op de "Help"-toets leest STedi de "helpfile" in buffer acht. De helptekst zit dus niet ingebouwd, wat het voordeel heeft dat het programma zo klein mogelijk blijft.
Met behulp van alternate-kommando's kan razendsnel van de ene buffer naar de andere worden overgeschakeld. Search-en replace-opdrachten werken helaas maar op één buffer tegelijk.

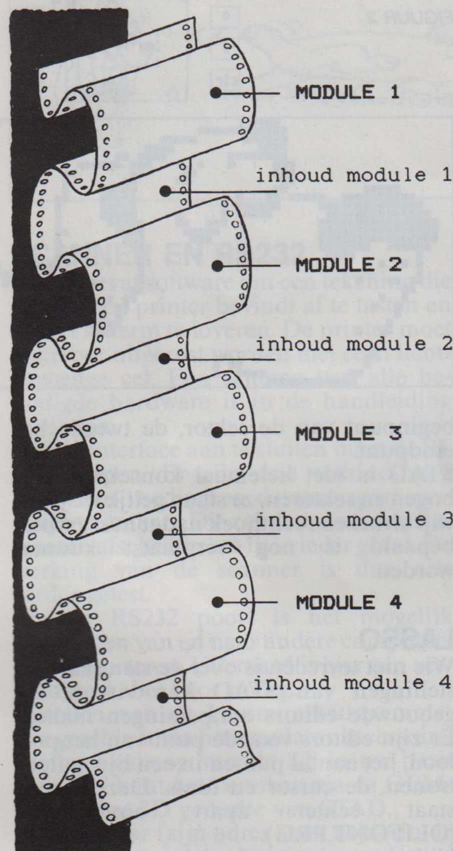
## FRAKTIES VAN SEKONDEN

STedi munt, zoals al eerder vermeld, uit in snelheid. Dit komt in allerlei dingen tot uiting. Bijvoorbeeld bij het scrollen van het scherm (dit doet STedi zelf, het operating system komt er niet aan te pas), file I/O (STedi gebruikt 8K blokken voor lees- en schrijfopdrachten; dit gaat veel sneller dan de analoge operaties via de desktop) en de search- en replace kommando's.
Enige cijfers: een search naar het woord 'testwoord', op de laatste regel van een 316.666 bytes grote, 4948 regels lange file, duurt ongeveer een halve sekonde. En een replace van het woord 'de', dat in deze tekst (op dit moment) 361 keer voorkomt, duurt minder dan een kwart sekonde! Dit is een grove schatting. Het gaat zo snel dat het met de stopwatch bijna niet is te meten. Voor de liefhebbers: uiteraard is Knuth bij de gebruikte algorithmes niet vergeten!
Om dit in een perspektief te plaatsen: First Word Plus doet over dezelfde replace zo'n 25 sekonden. Metingen met een ASCII-testfile (lengte 65 kByte) leverden de volgende cijfers op:
Inlezen: STedi ongeveer anderhalve sekonde, FirstWord Plus 20 sekonden.
Zoeken naar string 'testwoord' op het einde: STedi vrijwel momentaan, FirstWord Plus ongeveer 23 sekonden.
Ik denk dat deze cijfers aardig weergeven hoe extreem snel STedi is!

## UNIEK FOLDS-CONCEPT

STedi ondersteunt "folds" of vouwen. Ik zal proberen uit te leggen wat dit betekent.
Een programma bestaat meestal uit een aantal losse onderdelen: funkties, subroutines, deklaraties van variabelen, etc. Als

ieder los onderdeel van een file in een fold gestopt wordt is het mogelijk om met één kommando alle folds te sluiten. Van iedere fold blijft alleen de eerste regel zichtbaar. Op deze manier kan snel gezien worden hoe de struktuur van een file is. Vaak past een gevouwen file in zijn geheel op het scherm. Zoeken naar bepaalde stukken tekst behoort daardoor tot het verleden. De gebruiker opent de fold die hij nodig heeft, brengt zijn wijzigingen aan en sluit de fold weer.
Binnen folds kunnen overigens ook weer andere folds voorkomen. Deze nesting kan theoretisch oneindig diep zijn. De enige limiet is dat er niet meer dan 32767 regels in een fold mogen zitten.




## NIEUWE MUIS

Via de muis kan de gebruiker op een hele eigenzinnige manier kommando's geven. Zodra hij de muis aanraakt, wordt het pijltje op het scherm zichtbaar. De rechter muisknop maakt het mogelijk om door de tekst te scrollen, een fold te openen als hij dicht is of te sluiten als hij open is.
Met de linkerknop kan de kursor op iedere plaats in de file worden gezet en zijn kommando's mogelijk. Op het moment dat de linkerknop wordt ingedrukt, verandert de statusbalk namelijk van uiterlijk: op de plaats waar normaal kommando's worden ingetikt, verschijnt nu een aantal letters. Iedere letter staat voor een kommando dat bij aanklikken wordt uitgevoerd. Op deze manier kunnen met de muis blokoperaties, file-operaties (nu MET fileselector) en buffer-veranderingen worden uitgevoerd. Ook alle status-indicatoren zijn te veranderen door ze gewoon aan te klikken. Als op de filenaam wordt geklikt, laat STedi het komplete path zien (bijvoorbeeld f: \ start \ editor.art).
Het gaat te ver om alles uit te leggen, maar de integratie van de muis is duidelijk goed overdacht. Of het u aanspreekt of dat u liever de vertrouwde GEM-omgeving ziet, kan ik niet beoordelen. Zelf geef ik in ieder geval de voorkeur aan snelheid.

Naast de al genoemde eigenschappen heeft STedi nog veel meer in petto:

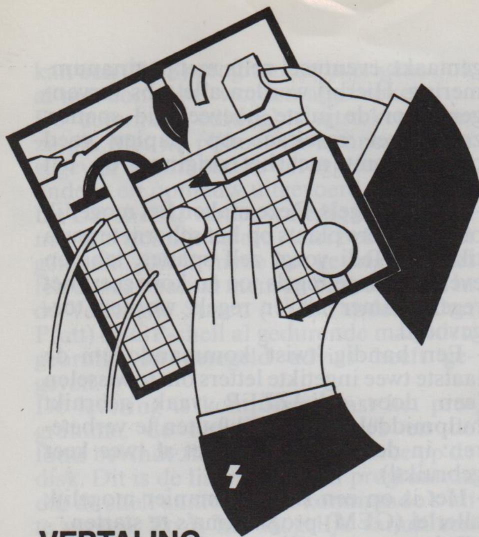
- Het is mogelijk om rechtstreeks kommando's aan een onderliggende shell te geven. Dit gaat op als STedi is aangeroepen vanuit een shell (bijvoorbeeld de GP-shell). De kommando's vanuit STedi komen dan direkt in de juiste routine van de shell terecht. De shell hoeft niet eerst van disk gelezen te worden. Hij staat immers nog in het geheugen.
- Naast de mogelijkheid om gewone blokken tekst te verplaatsen, te verwijderen of te kopiëren, is er ook de mogelijkheid om dit met kolommen te doen. Vooral bij het werken met tabellen is dit een zeer krachtig hulpmiddel.
- Een 'vind-het-volgende-woord' kommando. Dit is met name voor programmeurs erg makkelijk. De kursor staat op een bepaalde variabele en je wilt weten waar deze variabele verder voorkomt. Met een simpel "control"-kommando kan dan het volgende woord worden gevonden. Ook is het mogelijk het woord onder de kursor in de hele tekst te vervangen door een ander woord zonder het te vervangen woord in te tikken. Weinig gezien bij andere editors, maar erg handig!
- De aanwezigheid van zogenaamde "learn-buffers". Een learn-buffer is een soort recorder die alle toetsaanslagen onthoudt totdat hij wordt gestopt. Een vastgelegde learn-buffer kan zo vaak als gewenst worden afgespeeld en .... een learn-buffer mag andere learn-buffers bevatten! De learn-buffers worden bewaard in de "defaults-file", zodat ze slechts een keer hoeven worden ingetikt. STedi heeft tien learn-buggers.
- Auto-indent en word-wrap zijn aanwezig. Als auto-indent aanstaat, wordt iedere nieuwe regel precies begonnen onder de eerste niet-spatie op de voorgaande regel. Erg handig bij het laten inspringen van stukken programma (bijvoorbeeld tussen IF en THEN).
Word-wrap geeft STedi een klein vleugje tekstverwerker: als wordwrap aanstaat, wordt bij het bereiken van een op te geven kolom automatisch op de volgende regel verder gegaan. Voor het intikken van teksten als deze een onmisbare optie.
- Er kunnen andere dan de beide (gewone en kleine) karaktersets worden gebruikt. Ik gebruik bijvoorbeeld een IBM-achtig lettertype.
- STedi onthoudt de laatste acht kommando's die op de statusregel zijn ingetikt. Via "pijltje-omhoog" en "pijltje-omlaag" selekteert de gebruiker het juiste kommando. Eventueel kan hij de opdracht editten alvorens een return te geven. Dit is bijzonder handig bij zoek-kommando's en veel gebruikte file-namen.
- STedi werkt ook op een kleurenscherm, zij het dat altijd naar medium resolution wordt overgeschakeld. Bij het verlaten van de editor wordt alles uiteraard weer teruggezet, zodat de gebruiker in low resolution verder kan werken. Allerlei kleuren kunnen naar wens worden ingesteld en gesaved.
- Het is mogelijk om een binaire file te editten door hem in hex-mode in te lezen. Op deze manier kan STedi bijvoorbeeld gebruikt worden om in een bepaald programma een stuk tekst te veranderen.
- Krachtig tab-kommando. Ook onttabbing en her-tabbing is mogelijk.
- Spaties en tabs kunnen op het scherm zichtbaar gemaakt worden (met open respektievelijk dichte rondjes).
- Speciale printer-ondersteuning. Listings kunnen direkt vanuit de editor worden gemaakt; eventueel zelfs met paginanummering. Hierbij worden alle tabs vervangen door de juiste hoeveelheid spaties, zodat het resultaat op papier goed overeenkomt met het origineel op het scherm.
- Virtuele regels. Deze maken het mogelijk om op iedere plaats op het scherm te gaan tikken. STedi voegt zelf spaties, tabs en eventuele lege regels toe en korrigeert het regelnummer als er regels worden toegevoegd.
- Een handig 'twist' kommando om de laatste twee ingetikte letters om te wisselen (een door mij ZEER vaak gebruikt hulpmiddel om snel tikfouten te verbeteren: in deze zin heb ik het al twee keer gebruikt!).
- Het is op een simpele manier mogelijk allerlei (GEM) programma's te starten.
- Er is een eenvoudige manier om naar een bepaalde regel te springen: tik gewoon het regelnummer in op de statusregel. Dit lijkt triviaal, maar in First Word is het bijvoorbeeld niet mogelijk.
- Heel handig voor programmeurs: met een druk op de knop heeft u het haakje dat met het onder de kursor gelegen haakje korrespondeert te pakken. Dus: ga op een ( staan, geef het kommando, en de kursor springt naar het korresponderende ). Werkt ook met {} en [].

## WENSEN

Natuurlijk zijn er wel enige negatieve kanttekeningen te maken, zij het dat het hier om relatief kleine dingen gaat. Zo valt op de search toch nog wel wat aan te merken: 'wildcards' als een '*' of een '?' kunnen niet in een search gebruikt worden. Ook logische expressies zijn niet mogelijk (hierbij moet gedacht worden aan opdrachten als 'zoek een regel waarin het woord 'aap' wel, en het woord 'noot' niet voorkomt. Ik vraag me overigens af hoe vaak dergelijke samengestelde zoek-uitdrukkingen gebruikt zullen worden. Het zoek-kommando is zo snel dat men de selektie gemakkelijk met de hand kan doen.
Ook een splitscreen mode zou ik wel waarderen, alhoewel het natuurlijk mogelijk is met STedi een aantal files tegelijk te editten en dan snel van buffer te wisselen.
Omdat GEM afwezig is, worden kommando's niet via pull-down menu's gegeven, maar via het toetsenbord. Dit zal voor een aantal mensen een nadeel zijn, maar iedereen die vaak met GEM-programma's werkt, weet dat het lastig is om alle kommando's met de muis te geven. Het toetsenbord, dat toch al gebruikt wordt om de tekst in te tikken, is dan veel handiger. Een nadeel is natuurlijk dat het wat meer tijd kost om de kommando's te leren. Deze tijd zal echter snel genoeg weer zijn terugverdiend.
Een laatste minpuntje: ik mis een soort 'reformat' kommando om regels die door een uitbreiding de rechter kantlijn passeren, weer netjes binnen deze grens te passen. Nu heb ik een 'learn-buffer' moeten maken die een spatie aan het einde van een regel zet (en daarmee te lange regels automatisch afbreekt).

## ☆ KONKLUSIE

*STedi zal iedere gebruiker bevallen door de enorme snelheid en de overvloed aan slimme extra's. De (weinige) nadelen vallen mijns inziens ruimschoots weg tegen de voordelen. En voor de te verwachten prijs (ongeveer fl. 200,-) hoeft niemand het te laten.*



# •ST AIDED DESIGN•

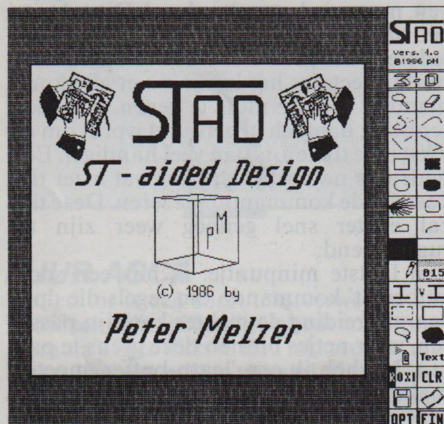
**Tekenpakketten voor de ST verdringen elkaar bij hun pogen een plaatsje in de vitrine van de Atari-dealer in de wacht te slepen. Om zich van de konkurrentie te onderscheiden is een duel in tekenfaciliteiten losgebarsten. STAD heeft het wat anders aangepakt: het is eigenlijk een geïntegreerd pakket. Naast tekeningen kunnen ook drie-dimensionale figuren gevormd worden. Tot voor kort moest de konsument die op beide terreinen aktief wilde zijn hier twee aparte programma's voor kopen. Cees van Nieuwenhuizen kijkt wat de som is van de delen.**

## VERTALING

STAD, een tekenpakket voor twee- en driemensionale figuren, biedt meer dan de firma Application Systems Heidelberg aanvankelijk dacht. De handleiding is zo dik geworden dat hij niet meer in de standaardverpakking past. Daarom is er voor de weinig elegante oplossing gekozen om hem te splitsen. Deel I van de handleiding zit in de doos, deel II is achter de doos meeverpakt. De inhoud aanpassen aan de verpakking: hoort dat niet andersom?
De tekst in het programma is in het Engels, maar de handleidingen zijn (nog) in het Duits. Dat is niet prettig, maar een logisch gevolg van de hausse die de ST in Duitsland doormaakt. Het is te hopen dat er ook een Nederlandse versie komt, maar dat is uiteraard afhankelijk van de belangstelling van de konsument. Voor vijftig exemplaren kun je tenslotte geen vertaler aan het werk zetten. Dus als het kopieercircuit zich nu eens koest houdt....

## SPUITEN MET VULPATRONEN

Nadat het pakket geladen is, staat het logo van STAD op het scherm (figuur 1). Rechts op het beeld is ook de menubalk al zichtbaar. Zo gauw je de muis beweegt wordt het scherm gewist en kan het tekenen beginnen. De muispointer wordt gesymboliseerd door een hand. Die hand zit opgesloten tussen de randen van de menubalk. Men moet een keuze maken voor het tekenvel betreden kan worden.




FIGUUR 1

Vanuit het tekenvel springt de gebruiker via de rechter muisknop weer terug naar het menu. De plaats waar men op het tekenvel aan het werk was, wordt echter onthouden. Een nieuwe keuze uit de menulijst voert rechtstreeks naar deze plaats terug.
De standaardmogelijkheden van lijnen, cirkels, rechthoeken, spuitbussen, etc. die we de laatste tijd in elk tekenpakket wel aantreffen zal ik hier niet bespreken. STAD bezit ze allemaal en ze funktioneren naar behoren. De spuitbus kan zelfs vulpatronen produceren en volgens zo'n patroon ook een tekening weer langzaam wissen.
Andere mogelijkheden waarmee STAD zich onderscheidt zijn het onderwerp van het vervolg van deze bespreking.

## KONVERTER

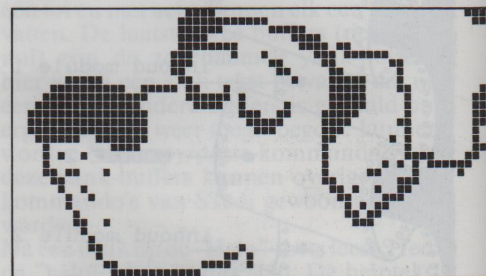
STAD maakt, afhankelijk van het beschikbare geheugen, maximaal gebruik van zestien beeldschermen. Hiervan is er één in gebruik als buffer voor kopieeroperaties. Door de schermen onderling te verwisselen kan een animatie-effekt bereikt worden.
Het is mogelijk om twee of vier beeldschermen in verkleinde vorm tegelijk op het scherm te projekteren.
Ook afdrukken van de schermen zijn op meerdere manieren mogelijk: de gewone screendump, vergroot en negentig graden gedraaid, twee onder elkaar (A4), twee klein en vier klein zijn hier de lekkernijen.
Naast het eigen STAD-formaat wordt bij het laden ook Degas, Doodle of Neochrome ondersteund. Voor zover het om lage resolutie gaat, konverteert STAD de 'gasten' automatisch.
Na het laden werd ik een aantal keren verrast door de melding "sorry, open error". Het raadselachtige was dat ik dan toch een tekening had binnengekregen.

## TAARTPUNT

Voor fijn tekenwerk is een loep onontbeerlijk. De versie van STAD is werkelijk fenomenaal. De gebruiker krijgt een rechthoekje op het scherm dat zelfs al een beetje vergroot terwijl je het over de tekening verplaatst. Na een druk op de linker muisknop wordt het gedeelte onder de loep op een apart scherm verder uitvergroot (zie figuur 2). Via de cursorpijltjes is de loep dan alsnog te verplaatsen. Nadeel is de vaste vergrotingsfaktor, maar de punten zijn fors genoeg om ze afzonderlijk aan te kunnen wijzen.
Uiteraard glijdt zelfs de beste tekenaar wel eens uit. Geen nood: STAD laat u zelf de grootte van uw vlakgum bepalen. Een paar pixels of een half scherm groot, het is aan u. Heel komfortabel.
Als bijzondere tekenoptie trof ik gebogen lijnstukken aan. De eerste muisklik bepaalt het beginpunt van een lijnstuk, de tweede het eindpunt en de derde de kromming.
Daarnaast zijn er sektoren. Een sektor is te vergelijken met een taartpunt. De vorm van de 'taart' kan zowel een cirkel als een ellips zijn. De eerste muisklik bepaalt het

FIGUUR 2

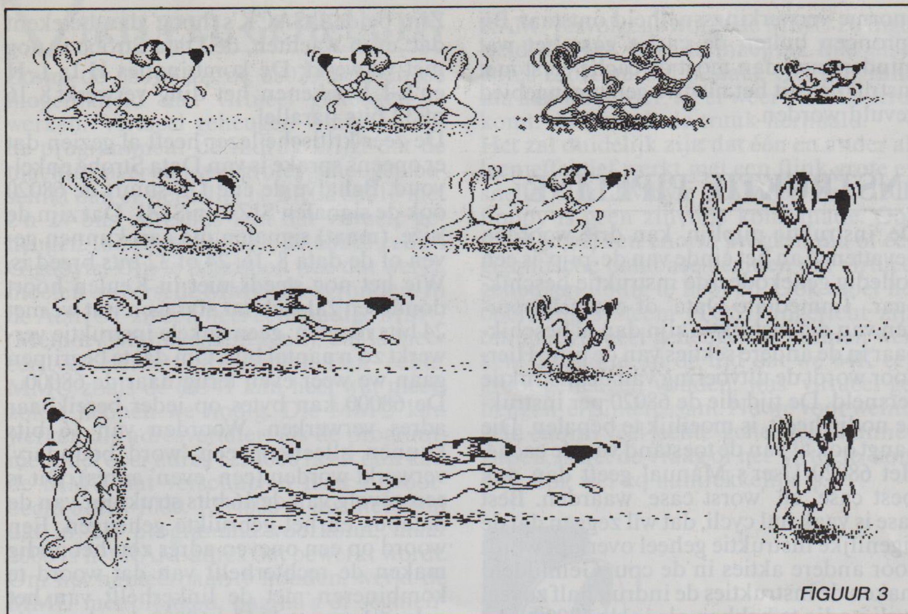


beginpunt van de sektor, de tweede het eindpunt.
STAD is niet helemaal konsekwent bij bogen en sektoren: ze staan gelijk vast, terwijl cirkel en rechthoek nadat hun grootte bepaald is nog verplaatst kunnen worden.

## LASSO

Wie niet tevreden is over de standaardinstellingen van STAD kan via de ingebouwde editors aanpassingen maken. Er zijn editors voor de punt van het potlood, het aantal punten in een lijn, vulpatronen, de cursor en fonts. Deze laatste staat echter apart op schijf (OLIFONT.PRG).
Uitknippen van onderdelen van tekeningen kan zoals gebruikelijk in de vorm van rechthoeken. Nieuw is de mogelijkheid om zelf de vorm van de uitsnede te bepalen: de lasso. Het komt bij het uitsnijden in de vorm van een blok vaak voor dat een deel van een aangrenzende tekening mee wordt uitgesneden. Dat is met de vrije uitsnede verleden tijd.
De losgesneden onderdelen komen in een aparte buffer terecht, van waaruit ze verder bewerkt kunnen worden. Alle mogelijkheden van spiegelen, vergroten en buigen zullen hier niet aan de orde komen. Figuur 3 toont enkele effekten. Alleen dient opgemerkt te worden dat draaien om een willekeurige hoek erg lastig is. Dit moet namelijk via een tussenbezoek aan het 3D gedeelte.
STAD start op met het standaard GEM-font. Alle bekende manipulaties (bold, italics, etc.) zijn mogelijk. Ook het formaat van de letters en de stand waarin ze op het scherm moeten komen is (in stappen van negentig graden) van te voren te bepalen. Bovendien kan ingesteld worden of de letter de ondergrond vervangt (replace), of dat alleen de zwarte punten van de lettermatrix worden neergezet (or).
Tenslotte wordt een zeventál fonts bijgeleverd dat ingeladen kan worden. STAD houdt er desgewenst vijf tegelijk in het geheugen vast.



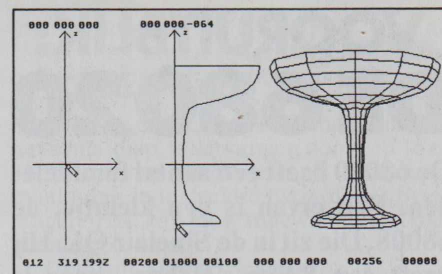
FIGUUR 3

## SCANNER EN RS232

STAD bevat software om een tekening die zich in een printer bevindt af te tasten en op het scherm te toveren. De printer moet daarvoor uitgerust worden met eeen lichtgevoelige cel. Een schema van alle benodigde hardware is in de handleiding te vinden.
Om de interface aan te sluiten moet de ST echter open. Er zijn drie verbindingen nodig om voeding en datatransport mogelijk te maken. Dit zal de meeste consumenten afschrikken. Mij in ieder geval. De werking van de scanner is dus niet verder getest.
Via de RS232 poort is het mogelijk tekeningen van en naar andere computers te transporteren. Uiteraard is hier een verbindingskabel voor nodig.
Instellen van de kommunikatieparameters geschiedt via het emulator accessoire, dat bij aankoop van de ST in uw bezit kwam. U kunt het bereiken via de GEM-balk in het 3D gedeelte van STAD.
Bij de auteur (zijn adres staat achterin de handleiding) is een terminalprogramma voor de C64 te verkrijgen dat prima met deze funktie van STAD samenwerkt.

## DE DERDE DIMENSIE

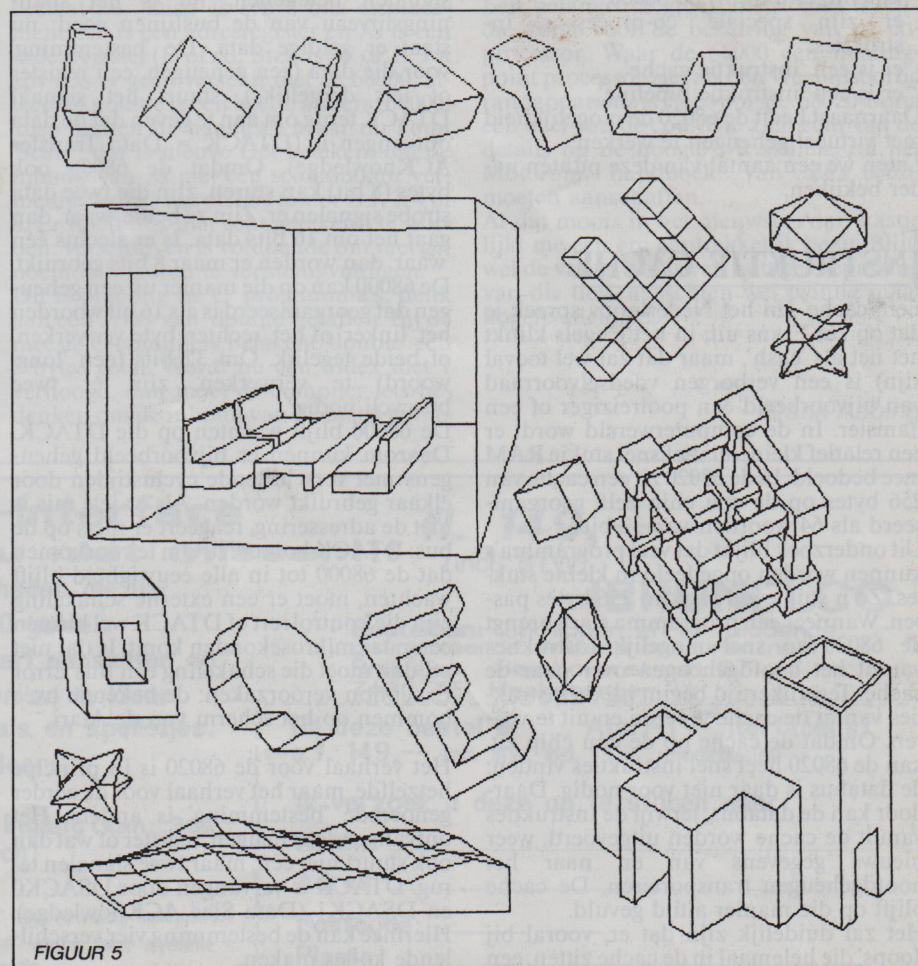
Het 3D gedeelte van STAD is via het tekenpakket bereikbaar. Daarbij kan een tekening meegenomen worden. Andersom kan een projektie van een 3D afbeelding naar het tekengedeelte getransporteerd worden, zodat alle mogelijkheden van beide helften van het programma uitwisselbaar zijn.
De vormgeving van het 3D gedeelte is wat bescheiden: een gewone GEM-balk. Wel even een teleurstelling als je de visuele presentatie van de tekenafdeling gewend bent.
Het 3D gedeelte laat zich het best vergelijken met het programma 3D-Frames dat in START 4 gepubliceerd werd. Uiteraard zijn de mogelijkheden van STAD vele malen groter, maar ook hier is een ontwerp te vergroten, te verkleinen, te draaien en te bewegen. Ontwerpen kunnen ook gekopieerd worden, zodat een bewegende kubus in een ommezien in vier bewegende kubussen verandert. De draaisnelheid van vier objekten tegelijk blijft heel behoorlijk.
Het is ook nog mogelijk om die vier kubussen tegelijkertijd van vier verschillende gezichtspunten te bekijken. En ook dan blijft alles draaien!
Het maken van een 3D ontwerp kan op twee manieren: door roteren (de manier van 3D-Frames) en via een draadmodel. Bij roteren wordt een gebogen lijn om de z-as gedraaid tot een volle figuur is verkregen (zie figuur 4).
Een draadmodel konstrueren is een langdurige bezigheid. Elk punt, lijn, vlak en de samenhang tussen de vlakken moet afzonderlijk opgegeven worden. Een kompleet ontwerp is uiteraard in alle mogelijke standen te bezichtigen. Voorbeelden van draadmodellen ziet u in figuur 5.




FIGUUR 4

*STAD is een pakket met bijzonder veel mogelijkheden. Wie zowel in tekenen als in 3D ontwerpen geïnteresseerd is, doet met het pakket een goede keus. De vormgeving van het 3D gedeelte zou echter nog wel een behandeling mogen krijgen. Door de keuze voor een GEM-balk komt STAD een beetje over als twee aan elkaar geplakte programma's en dat is jammer.*
*Het programma is duidelijk geschreven door een ST-fanaat. Dat is te zien aan de mogelijkheden "scanner" en "rs232". Niet voor iedereen nuttig en bruikbaar, maar er met plezier ook nog even aan toegevoegd. Dat spreekt aan, maar met veel gereedschap in een kist is het ook lang zoeken tot je het juiste gevonden hebt.*

*De handleiding van STAD is onontbeerlijk om alle mogelijkheden te ontdekken (er zit bijvoorbeeld nog van alles onder het toetsenbord verborgen). Daarom is het jammer dat deze nogal rommelig is. Het gedeelte "eerste kennismaking met STAD" staat bijvoorbeeld pas op bladzijde 112 in deel II. In deel I wordt elke optie besproken, maar een aantal belangrijke tips bij die opties staat weer in deel II. Kortom: heel wat bladeren.*



FIGUUR 5

# VOORUITBLIK: DE 68020

**De 68000 heeft een aantal familieleden. Eén ervan is een kleintje, de 68008. Die zit in de Sinclair QL. Hij heeft een 8-bits databus, maar is verder volledig compatibel met de 68000. De andere familieleden zijn groter. Dat zijn bijvoorbeeld de 68010 en de 68020. De huidige Atari ST bevat een 68000. In de machine die voor eind van dit jaar is aangekondigd, zal waarschijnlijk een 68020 zitten. Wat zijn de mogelijkheden van deze nieuwe chip? Frank Abbing blikt vooruit.**

## FEATURES

De 68020 kan alles wat de 68000 kan, en nog wat meer. Dat meerdere is een gevolg van een grotere instruktieset, een hogere snelheid en een verbeterde architektuur. Die hogere snelheid wordt veroorzaakt door een aantal faktoren:

- de kloksnelheid is 16 Mhz, in plaats van de standaard 8 Mhz van de 68000.
- sommige instrukties hebben minder cycli nodig.
- de databus kan 32 bits breed zijn: een 'long' wordt in één slag verwerkt.
- de adresseermogelijkheden zijn nog krachtiger dan bij de 68000.
- er zijn speciale 'co-processor' instrukties.
- er is een 'instruktie cache'.
- er is een 'instruktie pipeline'.

Daarnaast heeft de 68020 de mogelijkheid met virtueel geheugen te werken.
Laten we een aantal van deze punten nader bekijken.

## INSTRUKTIE CACHE

Een 'cache' (in het Nederlands spreek je dat op z'n Frans uit; in het Engels klinkt het net als 'cash', maar dat zal wel toeval zijn) is een verborgen voedselvoorraad van bijvoorbeeld een poolreiziger of een hamster. In de computerwereld wordt er een relatief klein, relatief snel stukje RAM mee bedoeld. In de 68020 zit een cache van 256 bytes op de cpu chip zelf, georganiseerd als 64 woorden van 32 bits.
Uit onderzoek blijkt dat veel programma's kunnen worden opgedeeld in kleine stukjes. Zo'n stukje zal vaak in 256 bytes passen. Wanneer een programma start, brengt de 68020 zo snel mogelijk instrukties vanuit het hoofdgeheugen over naar de cache. Tegelijkertijd begint hij de instrukties vanuit de cache te lezen en uit te voeren. Omdat de cache op de cpu chip zit, kan de 68020 heel snel instrukties vinden: de databus is daar niet voor nodig. Daardoor kan de databus, terwijl de instrukties vanuit de cache worden uitgevoerd, weer nieuwe gegevens van en naar het hoofdgeheugen transporteren. De cache blijft op die manier altijd gevuld.
Het zal duidelijk zijn dat er, vooral bij 'loops' die helemaal in de cache zitten, een enorme verwerkingssnelheid ontstaat. Bij sprongen buiten de cache gaat het wat minder snel: dan moet de cache eerst met instrukties uit het nieuwe geheugengebied gevuld worden.

## INSTRUKTIE PIPELINE

De 'instruktie pijplijn' kan drie woorden bevatten. Aan het einde van de 'pijp' is een volledig gedekodeerde instruktie beschikbaar. 'Immediate Data' of extensiewoorden van die instruktie zijn dan al beschikbaar in de andere sekties van de pijp. Hierdoor wordt de uitvoering van de instruktie versneld. De tijd die de 68020 per instruktie nodig heeft, is moeilijk te bepalen. Die hangt ook af van de toestand van de cache. Het 68020 User's Manual geeft dan ook 'best case' en 'worst case' waarden. Best case is vaak nul cycli, dat wil zeggen dat de eigenlijke instruktie geheel overlapt wordt door andere akties in de cpu. Gemiddeld maken de instrukties de indruk half zoveel cycli nodig te hebben als in de 68000. Dat zou, met de twee keer zo hoge kloksnelheid, de 68020 vier keer zo snel als de 68000 maken. Maar er zijn nog meer faktoren die de snelheid verhogen.

## DATABUS

Zoals bekend hebben de 68008 en de 68000 een 8- respektievelijk 16-bits databus. De 68020 kan zowel met een 8-, een 16- of een 32-bits databus werken. Om dat te verklaren, kijken we eerst even hoe de 68000 de bus behandelt.
De 68000 begint (bij schrijven) met het instellen van de buslijnen: hoog of laag. Even later stuurt de 68000 twee 'data strobe' signalen (UDS en LDS). Die twee signalen betekenen: nu is het spanningsniveau van de buslijnen goed: nu staat er geldige data. De 'bestemming' voor die data (een geheugen, een register of iets dergelijks), stuurt het signaal DTACK terug om aan te geven dat de data ontvangen is (DTACK = Data Transfer ACKnowledge). Omdat de 68000 ook bytes (8 bit) kan sturen, zijn die twee data strobe signalen er. Zijn ze beide 'waar' dan gaat het om 16 bits data. Is er slechts één 'waar' dan worden er maar 8 bits gebruikt. De 68000 kan op die manier uit een geheugen dat georganiseerd is als 16 bit woorden het 'linker' of het 'rechter' byte verwerken, of beide tegelijk. Om 32 bits (een 'long' woord) te verwerken, zijn er twee buscycli nodig.
De 68000 blijft wachten op die DTACK. Daarom kunnen er bijvoorbeeld geheugens met verschillende cyclustijden door elkaar gebruikt worden. Als er iets mis is met de adressering, reageert er niets op de bus: DTACK komt niet. Om te voorkomen dat de 68000 tot in alle eeuwigheid blijft wachten, moet er een externe schakeling zijn die kontroleert of DTACK wel binnen een paar mikrosekonden komt. Is dat niet zo, dan moet die schakeling een Bus Error Exception veroorzaken: de bekende twee bommen op het scherm van de Atari.

Het verhaal voor de 68020 is in principe hetzelfde, maar het verhaal voor de eerder genoemde 'bestemming' is anders. Het ontvangende geheugen, register of wat dan ook stuurt niet één, maar twee signalen terug. DTACK is vervangen door DSACK0 en DSACK1 (Data Size ACKnowledge). Hiermee kan de bestemming vier verschillende kodes maken. Zijn beide DSACK's 'hoog' dan betekent dat: even wachten, de Data Strobe is nog niet verwerkt. De kombinaties H-L, L-H en L-L betekenen: het 'ding' verwerkt 8, 16 of 32 bits parallel.
De zeer kritische lezer heeft al gezien dat er opeens sprake is van Data Strobe enkelvoud. Behalve de ene DS stuurt de 68020 ook de signalen SIZ0 en SIZ1. Dat zijn de 'size' (maat) signalen die aan kunnen geven of de data 8, 16, 24 of 32 bits breed is. Wie het nog steeds niet in Keulen hoort donderen zal zich nu afvragen wat we met 24 bits moeten: geen enkele instruktie verwerkt zo'n aantal bits. Om dat te begrijpen gaan we weer even terug naar de 68000.
De 68000 kan bytes op ieder bereikbaar adres verwerken. Woorden van 16 bits kunnen alleen op een 'word boundary' verwerkt worden (een 'even' adres). Dit is een gevolg van de 16 bits struktuur van de databus en het gebruikte geheugen. Een woord op een oneven adres zou het nodig maken de rechterhelft van dat woord te kombineren met de linkerhelft van het volgende woord. De 68000 hardware is daar niet op ingericht. Bovendien kost het tweemaal zoveel tijd: er zouden twee buscycli nodig zijn. Daarom maakt de 68000 een 'Address Error Exception' als je probeert een 16- of 32-bits woord een oneven adres te geven (drie bommen op de Atari ST).
In analogie hiermee mag verwacht worden dat de 68020 eist dat de adressen van 'long' woorden door vier deelbaar zijn. Dat is echter niet zo. De ontwerpers van de 68020 hebben hier goed over nagedacht. De enige beperking in de adressering is dat instrukties op een even adres moeten beginnen: data, hetzij byte, word of longword mag op ieder adres beginnen. Hoe kan dat werken?
Tussen de 32 databus-pennen van de 68020 en de interne cpu logika zit een multiplexer. Een multiplexer is eigenlijk een schakelaar met meerdere standen. Deze multiplexer mag worden opgevat als een schakelaar met vier standen en 32 sekties. De bedrading (voorzover je in een I.C. van bedrading mag spreken) zorgt ervoor dat de vier bytes uit de buitenwereld op vier manieren met de vier bytes op de interne 68020 bus verbonden kunnen worden. Wat 'buiten' de linker-byte is, kan dat 'binnen' ook zijn; dat is bij de multiplexer de stand 'rechtuit'. Maar het buiten-byte kan ook verbonden worden met een van de andere bytes.
Neem nu aan dat we een stuk geheugen hebben waarin op adres 100 een longword staat. Het volgende longword staat dan op adres 104, enzovoorts. Neem vervolgens aan dat de 68020 probeert een longword te lezen van adres 105. Door de 32 bits struktuur van het geheugen worden de bytes 104, 105, 106 en 107 gelezen, maar door de stand van de multiplexer worden de bytes 105 tot 107 links aangeschoven in de 68020 binnen gehaald. Dan moet er nog een geheugen cyclus volgen om byte 108 te lezen. Dat verschijnt links op de bus, maar de multiplexer staat nu zo dat het rechts in de 68020 komt. Alles komt dus waar het moet zijn; het kost alleen een extra slag. Het programma werkt daarom wel sneller als alle 'longs' op door vier deelbare adressen zitten!
Een iets andere taak van de multiplexer is te zorgen dat er ook met een 16- of 8-bits bus gewerkt kan worden. De twee DSACK signalen vertellen de 68020 dat er 1, 2 of 4 bytes verwerkt zijn. Daardoor kan de cpu zelf vast stellen hoeveel bytes er eventueel nog moeten volgen.

## VIRTUEEL GEHEUGEN

Zowel de 68010 als de 68020 hebben de mogelijkheid met virtueel geheugen te werken. 'Virtueel geheugen' houdt in dat de programmeur de indruk krijgt dat de machine over een veel groter geheugen beschikt dan er werkelijk is. Bij de 68010 met z'n 24 bits adresbus kan dat virtuele geheugen 16 Mbyte zijn; bij de 68020 zelfs 4 Gigabyte! Om te begrijpen hoe dat werkt, moeten we eerst de MMU doorgronden. In de Atari ST zit een heel eenvoudige MMU (Memory Management Unit), dat is meer een verbeterde chip-selector. Een 'echte' MMU zoals de 68541 kan heel wat meer; zoiets hebben we nodig. Die MMU kan werken als adresvertaler: als de programmeur het over adres 1234000 heeft dan kan de MMU dat verschuiven naar bijvoorbeeld adres 1000. 1234000 is dus het getal dat in het programma voorkomt, maar 1000 is het RAM-adres dat bewerkt wordt. Om het hanteerbaar te houden, werkt de MMU met blokken, pagina's of segmenten. (Drie woorden voor hetzelfde!) Adres 1234098 in het programma is dan, doorbordurend op het voorbeeld, 1098 in het RAM. Hoe groot een blok is, hangt af van de MMU instelling. Het zou best 1K of 100K kunnen zijn.

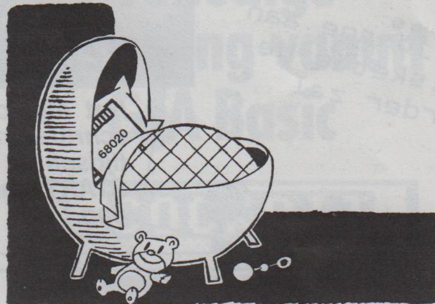
Een opmerking: bij oudere cpu's was een doel van de MMU meestal het uitbreiden van het adresbereik. Het systeem bevatte dan meer RAM dan de cpu zonder hulp kon adresseren. Dat is niet zo bij de 68541 en z'n soortgenoten: het adresbereik van de 68000 of 68020 is al groot genoeg zoals het is.

Terug naar de 68010/68020. Laten we aannemen dat de MMU zo is ingesteld dat het adresbereik van de machine 100K..200K is. Welke adressen dat werkelijk zijn, is voor ons als gebruiker minder belangrijk. Nemen we verder aan dat het programma verwijst naar adres 250K. Dat adres 'kent' de machine niet: er wordt een 'bus error' gegenereerd.

Even een zijstapje naar de Atari ST: wat betekenen daar twee bommen? Ja juist: bus error! Maar wanneer krijgen we die? Zoals eerder opgemerkt is er een externe schakeling die, bij uitblijven van DTACK, bus error maakt. In de Atari gebeurt dat als je een adres boven de 4M gebruikt, of als je een stukje beschermd geheugen adresseert. Uitblijven van een bus error betekent dus niet dat er echt geheugen op een bepaald adres zit!

Verder met het 68010/68020 verhaal: de routine die de bus error afhandelt, leest een stuk programma van de disk in het RAM. Vervolgens wordt de MMU zo ingesteld dat het voor de machine lijkt alsof adres 250K nu wel bestaat. Het programma kan nu verder, tot er weer een bus error komt: dan wordt de truuk herhaald.

Het zal duidelijk zijn dat één en ander alleen effektief werkt met een flink grote en snelle disk. 2Mbyte RAM en een 720K floppy is geen zinvolle kombinatie. Ook moet je wel een enorm programma of een gigantische database bezitten om virtueel geheugen nodig te hebben. Ook moet het operating systeem (TOS?) ingericht zijn om met virtueel geheugen te kunnen werken. Tenslotte: in de tijd dat we 64K een groot geheugen vonden, was 'virtueel geheugen' erg interessant. Nu we voor weinig geld enorm veel 'echte' geheugens kunnen kopen, is het de vraag of het 'virtuele' koncept nog wel zo aantrekkelijk is.




## INSTRUKTIESET

Dit stukje gaat ervan uit dat u al enigszins bent ingewijd in de instruktieset van de 68000. Bij 'grote broer' 68020 hebben de adresseermethodes wat uitbreidingen ondergaan:

(disp,An,Xi.Size*Scale)

bestaat nu. 'Disp' is een 8 bits displacement, An is een adresregister en Xi is een indexregister (D of A), 'Size' is .W of .L. Tot zover is er geen verschil met de 68000, alleen de notatie is een beetje anders (68000: 'disp' buiten de haakjes!) Maar nu komt 'Scale': dat is nieuw. Het betekent dat de inhoud van Xi met een schaalfaktor vermenigvuldigd wordt. Die faktor is 1, 2, 4 of 8. Er hoeft dus niet echt gerekend te worden: verschuiven is voldoende om met genoemde faktoren te vermenigvuldigen.

Dit is prachtig in 'C' programma's: denk aan een array van 'double' getallen. Bij de meeste C-compilers betekent 'double' een 8 bytes getal. Wordt nu een index met 1 verhoogd, dan moet de compiler iets bedenken om de inhoud van Xi met 8 te vermenigvuldigen. Een C-compiler voor de 68020 hoeft alleen maar de schaalfaktor op 8 te zetten. Niet alleen makkelijker, ook veel sneller!

Een andere uitbreiding is de mogelijkheid het 8 bits 'disp' te vervangen door een 16 of 32 bits 'base displacement'. Uiteraard maakt dat de instruktie twee of vier bytes langer. Het kan nog ingewikkelder, met 'memory indirect':

([bd,An],Xi.Size*Scale,od)

Hierin is 'bd' het genoemde base displacement en 'od' het 'outer displacement'. De werking is als volgt: eerst telt de 68020 bd en An bij elkaar. Het resultaat wordt gebruikt als adres van een longword. Dit longword wordt gelezen en bevat het indirekte adres. Daar wordt Xi bijgeteld, eventueel vermenigvuldigd met Scale, en het od. Het resultaat is het effektieve adres.

Het kan ook nog zo:

([bd,An,Xi.Size*Scale],od)

Eerst wordt het deel tussen rechte haken uitgewerkt. Het resultaat is het adres van een longword. De 68020 berekent het effektieve adres door het od op te tellen bij het longword.

68000 programmeurs zullen verwachten dat bovenbeschreven kunstjes in plaats van met An ook met de PC kunnen. Dat is ook zo. Alles bij elkaar een aantal mogelijkheden waarvan het gebruik misschien niet direkt duidelijk is. Eén van de dingen waar je aan zou kunnen denken, is een array van pointers in 'C'. Als deze nieuwe mogelijkheden door de C-compiler gebruikt worden, zorgen ze, afhankelijk van de programmastruktuur, nog eens voor een flinke snelheidswinst. Moeilijk te zeggen hoeveel. Ik schat een faktor twee.

De instruktieset is zelf ook flink verbeterd. Zo zijn er nu de 'long*long' vermenigvuldiging en de 'double/long' divide.

Een belangrijke groep instrukties is 'cp': die zorgt voor de besturing van de coprocessor. Waar de 68000 een floating-point processor ziet als een vreemdsoortig randapparaat, is het voor de 68020 bijna een deel van de cpu. Wie zich echt van de details op de hoogte wil stellen, zal het Motorola handboek van de 68020 moeten aanschaffen.

Al dat moois in het nieuwste Atari-kastje lijkt me een erg aantrekkelijk bezit. Blijft wel de vraag: wat is een nuttige toepassing van die tien, misschien wel twintig maal hogere snelheid...

*Frank Abbing*

# COCKPIT

**Veel ST-gebruikers werken bijna alleen nog met GfA Basic. Daar kleeft een nadeel aan: GfA Basic werkt niet volledig onder GEM. Allerlei zaken die zo handig via de desktop kunnen, zijn vanuit GfA moeilijk bereikbaar. Met COCKPIT heeft Rob Stroop een hulpprogramma geschreven dat voorkomt dat u Basic moet verlaten als u een desktop-functie wil aktiveren. Daarnaast zijn er nog vele handige funkties van eigen makelij toegevoegd.**
**Belangrijk: Cockpit is geschikt voor ALLE resoluties.**

## DEJA VU

Hoe vaak is het u niet overkomen: u werkt aan een programma dat hulpbestanden gebruikt en zou o zo graag de listing van het hulpbestand op het scherm zien. Maar dit betekent dat u wel even bezig bent: hoofdprogramma saven, GfA Basic verlaten, twee keer op het hulpbestand klikken, 'show' aktiveren, bekijken, en GfA en het hoofdprogramma weer laden.
Ook zult u zich de keer nog wel herinneren dat u geen geformatteerd schijfje bij de hand had om iets op te bergen. Of de keer dat de kleuren niet goed stonden. Of de muis. Of de toetssnelheid.

## TEGELIJK LADEN

Cockpit is een GfA procedure die u voortaan aan het begin van elke programmeer-aktiviteit inlaadt. Dat gaat erg eenvoudig: u kopieert GfA Basic en cockpit op een aparte schijf en kiest 'install application'. Daardoor komt er een formulier op het scherm, waarin u op de plaats van de kursor de extensie BAS invoert. Na een klik op 'ok' kiest u 'save desktop'. Alle instellingen worden hierdoor in het bestand 'desktop.inf' bewaard. Voortaan gebruikt u deze schijf om de ST op te starten als u in Basic wilt programmeren.
Als u twee keer op cockpit.bas klikt, wordt voortaan zowel GfA Basic als cockpit geladen. Via control-Z belandt u aan het einde van de cockpit-listing. Vanaf deze plek voert u uw eigen programma in. Op elk moment dat u een funktie van cockpit nodig heeft, drukt u op 'esc'. U komt daardoor in 'direct mode'. Daarin tikt u 'gosub cockpit', waarna het afgebeelde paneel op het scherm komt.

## GfA TOOLKIT

Van boven naar beneden biedt het paneel de volgende mogelijkheden:
***CLOSEBOX*** - de manier om het paneel te laten verdwijnen en weer terug te keren in de GfA interpreter.
***DATUM*** - klik op het cijfer dat u wilt veranderen. Links klikken verhoogt het cijfer cyclisch; rechts gaat het de andere kant op. Let er op dat u een bestaande datum invoert: 31.02.1987 wordt niet geaccepteerd.
***TIJD*** - van hetzelfde laken een pak. Ook hier alleen bestaande tijden invoeren.
***MORE*** -de manier om een ASCII-bestand op het scherm af te drukken. Na een aantal regels stopt de listing. U heeft dan de keuze tussen vervolgen met de spatiebalk (of een return) en stoppen via de letter Q(uit).
***COPY*** - de manier om een bestand (of een aantal bestanden) van de ene naar de andere schijf over te brengen. Selekteren gaat via de linker muistoets. Een vergissing kan met rechts ongedaan gemaakt worden. Eventueel kunt u de bestanden tegelijk in een nieuwe folder plaatsen. Handig als u een kopie op dezelfde schijf wilt hebben!
***FORMAT*** - deze optie bezorgt u een enkel- of dubbelzijdig geformatteerd schijfje. Formatteren gaat alleen via drive A.

## Volledige besturing vanuit GfA Basic




***NEW FOLDER*** - hiermee kan de gebruiker een folder met een vrij op te geven naam maken.
***EXEC*** (execute) - cockpit maakt het zelfs mogelijk om tussentijds andere programma's (PRG-bestanden) uit te voeren. Dit gaat vaak, maar niet altijd goed. Soms ontstaan er problemen in verband met verschillende resoluties. Andere programma's hebben een kursor nodig. U vindt in de listing (Procedure Executer) hoe u cockpit zo'n kursor kunt laten maken. Al experimenterende komt u er vanzelf achter welke programma's te gebruiken zijn.
Na het beëindigen van het programma keert u weer in de cockpit terug.
Cockpit gaat ervan uit dat de programma's zich op ramdisk D bevinden. Uiteraard is dit eenvoudig te wijzigen.
***DELETE*** - de manier om een bestand van schijf te verwijderen.
***COMPILER*** - wanneer u de GfA Basic compiler in uw bezit heeft, kunt u hem via de cockpit aktiveren. De compiler moet daarvoor wel op de boot-schijf staan.
***KEY REP*** - hiermee is in te stellen hoe snel de toetsen repeteren.
***KEY TAK*** - de aanvullende funktie: het getal geeft aan na hoeveel tijd een ingedrukte toets begint te repeteren.
***DEL.FOLDER*** - verwijdert een overbodige folder van schijf. Voor de naam van de folder moet de drive gespecificeerd worden.
De te verwijderen folder moet leeg zijn. Gebruik DELETE als dit nog niet het geval is.
***MEM*** - berekent het vrije ram-geheugen.
***DFRE*** - berekent de vrije ruimte op een schijf.
***WAARDE*** - de RGB-waarde van de kleur die in het onderste kleurvlak te zien is. Door links of rechts op de cijfers te klikken, is de kleur te wijzigen.
***INDEX*** - het nummer van de kleur die in het onderste kleurvlak getoond wordt. Het cijfer verandert door op een van de kleine vierkante kleurvlakjes te klikken. Deze kleur zal meteen in het grote vlak afgebeeld worden en is nu gereed om via WAARDE bewerkt te worden.

Let op: cockpit bestaat zelf ook uit 'kleuren'. U kunt het paneel hiermee onzichtbaar maken!

## ASCII-VERSIE

Cockpit ondersteunt twee diskdrives (A en B) en een ramdisk (D). Wanneer u liever een ramdisk onder een ander ikoon gebruikt, moet u de letter 'D' in de listing door een andere vervangen.
Voor sommige toepassingen maakt cockpit hulpbestanden aan (DIR.FOL, DIR.FLE). Na afloop kunt u ze met DELETE verwijderen.
Aan het einde van een sessie kunt u cockpit eenvoudig met een 'block delete' verwijderen alvorens uw programma te saven. Dit is eigenlijk alleen nodig als uw programma af is. Bij 'tussenstops' bewaart u programma en cockpit gewoon in hetzelfde bestand.
Het is verstandig om ook een schijf met een ASCII-listing van cockpit bij de hand te houden. Wie cockpit 'vergeet' te laden, kan hem op elk gewenst moment alsnog via 'merge' aan zijn programma toevoegen.

*Rob Stroop*

### AANWIJZINGEN BIJ DE LISTING

*Om ruimte te winnen is de listing in kolommen afgedrukt. Wanneer u in de listing het symbool => tegenkomt, is sprake van een afgebroken regel.*
*U moet het symbool => zelf niet intikken en ook niet op return drukken; maar gewoon verder gaan met de volgende regel.*

```
' *************************************
Procedure Cockpit
  Cls
  Deffn Curs_pos.kol=Int(Mousex/8)+1
  Deffn Curs_pos.reg=Int(Mousey/8)+1
  If Xbios(4)=2
    Fontptr$=Mki$(&HA000)+Mki$(&H2009 =>
)+Mki$(&H4E75)
    Lnavar$=Mki$(&HA000)+Mki$(&H4E75)
    Fontptr%=Varptr(Fontptr$)
    Fontptr%=C:Fontptr%()
    Lnavar%=Varptr(Lnavar$)
    Lnavar%=C:Lnavar%()
    H8x8=Lpeek(Fontptr%+4)
    Lpoke Lnavar%-22,Lpeek(H8x8+76)
    Dpoke Lnavar%-46,8
    Dpoke Lnavar%-40,640
    Dpoke Lnavar%-42,49
  Endif
  Deffn Med_resolution=Xbios(5,L:-1,L =>
:-1,1)
  Deffn Low_resolution=Xbios(5,L:-1,L =>
:-1,0)
  Deffn Present_resolution=Xbios(4)
  If @Present_resolution=0
    Flag1=1
  Endif
  @Tek_cock
  @Text_cockpit
  @Muis_handle
Return
' *************************************
Procedure Tek_cock
  X=47
  Y=13
  Box X,Y,X+80,Y+169
  Box X,Y,X+81,Y+170
  Line X+17,Y+2,X+72,Y+2
  Line X,Y+10,X+80,Y+10
  Line X,Y+18,X+80,Y+18
  Line X,Y+122,X+80,Y+122
  Box X+2,Y+157,X+78,Y+167
  Teller=0
  Kleur=0
  X=56
  Y=152
  While Teller<2
    If Kleur=0
      Color 1
      Box X,Y,X+7,Y+7
    Else
      Deffill Kleur,1
      Pbox X,Y,X+7,Y+7
    Endif
    X=X+8
    If X=120
      X=56
      Y=Y+8
      Teller=Teller+1
      Kleur=-1
    Endif
    Kleur=Kleur+2
  Wend
  Deffill 3,2,4
  Fill 55,25
  Deffill 2,2,6
  Fill 64,90
  Deffill 4,2,2
  Fill 50,161
  Default=Xbios(7,0,-1) And &H777
  Print At(7,3);
  Out 5,7
  Flag_0=1
Return
' *************************************
Procedure Text_cockpit
  Print Chr$(27);"p"
  Print At(9,3);"COCKPIT"
  Print At(7,5);Date$
  Print At(7,6);Spc(1);Time$;Spc(1)
  Print At(7,7);"**";Spc(1);"MORE";Sp =>
c(1);"**"
  Print At(7,8);"**";Spc(1);"COPY";Sp =>
c(1);"**"
  Print At(7,9);"*";Spc(1);"FORMAT";S =>
pc(1);"*"
  Print At(7,10);"NEW FOLDER"
  Print At(7,11);"**";Spc(1);"EXEC";S =>
pc(1);"**"
  Print At(7,12);"*";Spc(1);"DELETE"; =>
Spc(1);"*"
  Print At(7,13);"*COMPILER*"
  @Key.board_init
  Print At(7,14);"KEY REP*";Spc(2)
  Print At(7,15);"KEY TAK*";Spc(2)
  Print At(15,14);Rep%
  Print At(15,15);Tak%
  Print At(7,16);"DEL.FOLDER"
  Print At(7,17);"*";"MEM";"*";"DFRE*"
  Print At(7,18);
  Out 5,3
  Print At(8,18);"WAARDE";Hex$(Default)
  Print At(7,19);Spc(1)
  Print At(8,19);
  Out 5,2
  Print At(9,19);"INDEX 00"
  Deffill ,1,1
Return
' *************************************
Procedure Muis_handle
  While 1
    Vsync
    Print At(7,6);Chr$(27);"p";Spc(1) =>
;Time$;Spc(1)
    X=@Curs_pos.kol
    Y=@Curs_pos.reg
    If Mousek=1 And X=7 And Y=3
      Print Chr$(27);"q"
      Edit
    Endif
    If Mousek=1 And X>6 And X<17 And Y=9
      @Format
    Endif
    If Mousek=1 And X>6 And X<17 And =>
Y=10
      @New_folder
    Endif
    If Mousek=1 And X>6 And X<17 And Y=7
      @More
    Endif
    If Mousek=1 And X>6 And X<17 And =>
Y=12
      @Delete
    Endif
    If Mousek=1 And X>6 And X<17 And Y=8
      @Copy
    Endif
    If Mousek=1 And X>6 And X<17 And =>
Y=11
      @E.xec
    Endif
    If Mousek=1 And X>6 And X<17 And =>
Y=16
      @Delete_folder
    Endif
    If Mousek=1 And X>7 And X<11 And =>
Y=17
      @Memory
    Endif
    If Mousek=1 And X>11 And X<16 And =>
Y=17
      @D.free
    Endif
    If Mousek>0 And X>6 And X<17 And =>
Y=13
      @Compile
    Endif
    If Mousek>0 And X>6 And X<17 And =>
Y=14 Or Y=15
      @Key.board_init
      @Set_keyboard
    Endif
    If Mousek=1 And X>7 And X<16 And =>
Y>19 And Y<22
      @Get_index
      Box 49,170,125,180
      If @Present_resolution=0
        @Debug_col.index
      Endif
      @Get_kl.waarde
      Deffill Index,1,1
      Fill 55,174
    Endif
    If Mousek=1 Or Mousek=2 And X>13 =>
And X<17 And Y=18
      Box 49,170,125,180
      @Set_color
      Fill 55,174
    Endif
    If Mousek=1 Or Mousek=2 And X>7 A =>
nd X<16 And Y=6
      @Set_time
    Endif
    If Mousek=1 Or Mousek=2 And X>6 A =>
nd X<17 And Y=5
      @Set_date
    Endif
    Pause 10
  Wend
Return
' *************************************
Procedure Get_kl.waarde
  Kleur=Xbios(7,N.index,-1) And &H777
  Kleur$=Hex$(Kleur)
  If Len(Kleur$)=1
    Kleur$="00"+Kleur$
  Endif
  If Len(Kleur$)=2
    Kleur$="0"+Kleur$
  Endif
  Print At(14,18);Kleur$
Return
' *************************************
Rem DE WAARDE ,DIE MEN DOOR DE FUNCTI =>
E POINT() TERUG KRIJGT,IS NIET
Rem SYNCHROOM MET DE INDEX VAN DEZE K =>
LEUR.DIT IS EEN GFA_BUG.ZOLANG DEZE NOG
Rem NIET VERBETERD IS,IS DE ONDERSTAA =>
NDE PROCEDURE NODIG.BIJ EEN VOLGENDE =>
VERSIE
Rem VAN GFA_BASIC KUNT U DEZE PROCEDU =>
RE (HOPELIJK) WEGLATEN.TEVENS MOET U
Rem DAN DE REGELS SCHRAPPEN WAAR DEZE =>
 FUNCTIE AANGEROEPEN WORDT.(VERSIE 2.0)
Procedure Debug_col.index
  If Index=1
    N.index=Index+14
  Endif
  If Index=2 Or Index=3 Or Index>7 An =>
d Index<12
    N.index=Index-1
  Endif
  If Index=4 Or Index=12 Or Index=0
    N.index=Index
  Endif
  If Index=5 Or Index=13
    N.index=Index+1
  Endif
  If Index=6 Or Index=14
    N.index=Index-3
  Endif
  If Index=7 Or Index=15
    N.index=Index-2
  Endif
Return
' *************************************
Procedure Get_index
  Print At(14,19);
  Index=Point(Mousex,Mousey)
  If Index<10
    Print Spc(1);"0";Index;
  Else
    Print Spc(1);Index;
  Endif
Return
' *************************************
Procedure Set_color
  If Flag_0=1
    Kleur$="777"
  Endif
  Tel.=Val(Mid$(Kleur$,X-13,1))
  @Dec_inc
  If Tel.>7
    Tel.=0
  Endif
  If Tel.<0
    Tel.=7
  Endif
  Mid$(Kleur$,X-13,1)=Str$(Tel.)
  Print At(14,18);Kleur$
  K.leur$="&h"+Kleur$
  Setcolor N.index,Val(K.leur$)
  Flag_0=0
Return
' *************************************
Procedure Set_time
  T.ime$=Time$
  Tel.=Val(Mid$(T.ime$,X-7,1))
  @Dec_inc
  If Tel.<0
    Tel.=9
  Endif
  If Tel.>10
    Tel.=0
  Endif
  Mid$(T.ime$,X-7,1)=Str$(Tel.)
  Settime T.ime$,""
Return
' *************************************
Procedure Set_date
  D.ate$=Date$
  Tel.=Val(Mid$(D.ate$,X-6,1))
  @Dec_inc
  Mid$(D.ate$,X-6,1)=Str$(Tel.)
  Settime "",D.ate$
  Print At(7,5);Date$
Return
' *************************************
Procedure Dec_inc
  If Mousek=1
    Inc Tel.
  Endif
  If Mousek=2
    Dec Tel.
  Endif
  Dummy=0
Return
' *************************************
Procedure More
  Setcolor 3,0
  On Error Gosub More_errors
  @Save_cock
  Fileselect "\*.*","",File_name$
```

```
  @Set_resolution
  If File_name$>"" And File_name$<>"\"
    Open "U",#1,File_name$
    @Page_more
    Repeat
      Print Chr$(27);"p";
      Print " More ";
      Print Chr$(27);"q";
      Showm
      Repeat
        Toets=Inp(2)
      Until Toets=32 Or Toets=13 Or T =>
oets=113 Or Toets=81
      If Toets=32 Or Toets=13 Or Toet =>
s=113 Or Toets=81
        Rem toets 32=space_bar :toets =>
 13=CR:TOETS 81 ="Q" TOETS 113=q
        If Toets=32
          @Page_more
        Endif
        If Toets=13 And Eof(#1)=0
          Print Chr$(27);"M";
          Line Input #1,More$
          Print More$
        Endif
      Endif
    Until Eof(#1)<>0 Or Toets=113 Or  =>
Toets=81
    If Toets=81 Or Toets=113
      Goto Eop.more
    Endif
    Print Chr$(27);"M";
    Print Chr$(27);"p";
    Print " End of file ";''File_name$
    Print Chr$(27);"q";
    Void Inp(2)
    Eop.more:
  Endif
  @Reset_resolution
  Close
  @Load_cock
Return
' ***************************************
Procedure Page_more
  I%=0
  Print Chr$(27);"M";
  While I%<23 And Eof(#1)=0
    Line Input #1,More$
    Print More$
    Inc I%
  Wend
Return
' ***************************************
Procedure Load_cock
  Cls
  Showm
  Put 47,13,Cockp.$
  Print Chr$(27);"p"
Return
' ***************************************
Procedure Save_cock
  Hidem
  Print Chr$(27);"q"
  Get 47,13,128,183,Cockp.$
  Cls
  Showm
Return
' ***************************************
Procedure Copy
  Setcolor 3,0
  @Save_cock
  C.opy%=0
  Disk$="ABD"
  Extension$="\*.*"
  B.oxtext$="|"+"SOURCE-DISK"
  Alert 1,B.oxtext$,3,"A|B|D",Disk%
  @Set_resolution
  D.isk$=Mid$(Disk$,Disk%,1)+":"+Exte =>
nsion$
  Files D.isk$ To "DIR.FLE"
  Open "U",#1,"DIR.FLE"
  @Copy_init
  Close
  Erase File$()
  Erase Filenr.()
  @Reset_resolution
  Folder%=0
  C.ontinue=0
  C.opy%=0
  S.top=0
  @Load_cock
Return
' ***************************************
Procedure Copy_init
  Cls
  S.top=0
  File_aantal%=0
  Limit%=20
  Y.teller%=0
  While C.ontinue=0
    @Print_files
    @Print_disk.options
    @Select_files
    If S.top=1 And X>22 And X<31
      C.ontinue=1
    Endif
    If S.top=1 And X<10
      Limit%=Limit%+20
      S.top=0
      Cls
      Y.teller%=Y.teller%+20
      Print At(1,1);Chr$(27);"q";
    Endif
    If S.top=1 And X>13 And X<22
      If Delete%=0 And C.ontinue=0
        @Copy_executer
        C.ontinue=1
        C.opy%=0
      Endif
      If Delete%=1
        @Delete_executer
        C.ontinue=1
      Endif
    Endif
  Wend
  C.ontinue=1
Return
' ***************************************
Procedure Print_files
  On Error Gosub Print_files.error
  While Eof(#1)=0 And File_aantal%<Li =>
mit%
    D.i.m.error$=File$(0)
    Line Input #1,File$(File_aantal%)
    Print File$(File_aantal%)
    Inc File_aantal%
  Wend
Return
' ***************************************
Procedure Print_files.error
  If Err=15
    Dim File$(100),Filenr.(100)
    File$(0)=D.i.m.error$
    Resume
  Endif
Return
' ***************************************
Procedure Select_files
  While S.top=0
    While Mousek=1
      X=@Curs_pos.kol
      Y=@Curs_pos.reg
      Y%=Y+Y.teller%
      If Y=24
        S.top=1
        Goto S.top
      Endif
      If Mid$(File$(Y%-1),1,1)="*"
        @Folder_copy
        S.top=1
        C.ontinue=1
        Goto S.top
      Endif
      If Filenr.(Y%-1)=0
        Filenr.(Y%-1)=1
        Print At(1,Y);Chr$(27);"p";Fi =>
le$(Y%-1)
      Endif
      S.top:
    Wend
    While Mousek=2
      X=@Curs_pos.kol
      Y=@Curs_pos.reg
      Y%=Y+Y.teller%
      If Filenr.(Y%-1)=1
        Filenr.(Y%-1)=0
        Print At(1,Y);Chr$(27);"q";Fi =>
le$(Y%-1)
      Endif
    Wend
  Wend
Return
' ***************************************
Procedure Print_disk.options
  If Eof(#1)=0 And Delete%=0
    Print At(1,24);Chr$(27);"p";Spc(1 =>
);"CONTINUE";Chr$(27);"q";Spc(5);
    Print ;Chr$(27);"p";Spc(1);"COPY" =>
;Chr$(27);"q";Spc(5)Chr$(27);"p";"EX =>
IT";
    Print ;Chr$(27);"q"
  Endif
  If Eof(#1)<>0 And Delete%=0
    Print At(14,24);Chr$(27);"p";Spc( =>
1);"COPY";Spc(1);Chr$(27);"q";Spc(5);
    Print Chr$(27);"p";Spc(1);"EXIT"; =>
Chr$(27);"q";Chr$(27);"H";
  Endif
  If Eof(#1)=0 And Delete%=1
    Print At(1,24);Chr$(27);"p";Spc(1 =>
);"CONTINUE";Chr$(27);"q";Spc(5);
    Print ;Chr$(27);"p";Spc(1);"DELET =>
E";Chr$(27);"q";Spc(3);Chr$(27);"p";" =>
EXIT";
    Print ;Chr$(27);"q"
  Endif
  If Eof(#1)<>0 And Delete%=1
    Print At(14,24);Chr$(27);"p";Spc( =>
1);"DELETE";Chr$(27);"q";Spc(2);
    Print Chr$(27);"p";Spc(1);"EXIT"; =>
Chr$(27);"q";Chr$(27);"H";
  Endif
Return
' ***************************************
Procedure Folder_copy
  Positie%=Instr(1,File$(Y%-1),Chr$(32))
  Mid$(File$(Y%-1),1,1)="\"
  Mid$(File$(Y%-1),Positie%)=Extension$
  D.isk$=Mid$(Disk$,Disk%,1)+":"+Mid$ =>
(File$(Y%-1),1,17)
  Files D.isk$ To "DIR.FOL"
  Positie%=Instr(1,File$(Y%-1),Chr$(42))
  Folder$=Mid$(D.isk$,3,Positie%-2)
  Folder%=1
  Erase File$()
  Erase Filenr.()
  Close
  Open "U",#1,"DIR.FOL"
  @Copy_init
  C.ontinue=1
Return
' ***************************************
Procedure Copy_executer
  On Error Gosub Copy_errors
  Alert 1,"|DESTINATION DRIVE",1,"A|B =>
|D",Dest%
  Alert 1,"|MOETEN DE TE COPIEREN FIL =>
ES|IN EEN FOLDER",2,"JA|NEE|EXIT",D.v =>
raag%
  Disk.dest$=Mid$(Disk$,Dest%,1)+":"
  If D.vraag%=2
    Dest.folder$=""
    Disk.dest$=Disk.dest$+"\"
  Endif
  If D.vraag%=3
    Goto Eop.copy_executer
  Endif
  If D.vraag%=1
    If Folder%=1
      Alert 1,"|WELKE FOLDER",1,"SOUR =>
CE|ANDERE",Dest._folder%
    Else
      Goto Folder.naam
    Endif
    If Dest._folder%=1
      Dest.folder$=Folder$+"\"
      Print Dest.folder$
      D.folder$=Disk.dest$+Folder$
      Mkdir D.folder$
    Endif
    If Dest._folder%=2
      Folder.naam:
      Cls
      Print At(5,10);"NAAM FOLDER"
      Print At(17,10);
      Input "",Copy.folder$
      D.folder$=Disk.dest$+"\"+Copy.f =>
older$
      Mkdir D.folder$
      Dest.folder$="\"+Copy.folder$+"\"
    Endif
  Endif
  Start.of.copy:
  For I%=0 To File_aantal%
    Buff.size=8*1024
    Buff.$=Space$(Buff.size)
    If Filenr.(I%)>0
      Source$=Mid$(D.isk$,1,3)+Mid$(F =>
ile$(I%),2,13)+Chr$(0)
      If Folder%=1
        Source$=Mid$(D.isk$,1,2)+Fold =>
er$+"\"+Mid$(File$(I%),2,13)+Chr$(0)
      Endif
      Dest$=Disk.dest$+Dest.folder$+M =>
id$(File$(I%),2,13)+Chr$(0)
      Rem TRY TO OPEN SOURCE-FILE
      S.handle=Gemdos(&H3D,L:Varptr(S =>
ource$),0)
      If S.handle<0
        Print "CAN'T OPEN SOURCE-FILE"
        Goto Eop.copy
      Endif
      Rem TRY TO OPEN DESTINATION-FILE
      D.handle=Gemdos(&H3C,L:Varptr(D =>
est$),0)
      If D.handle<0
        Print "CAN'T OPEN DESTINATION =>
-FILE"
        Goto Label
      Endif
      Rem COPY SOURCE-FILE TO DESTINA =>
TION-FILE
      Repeat
```

```
      R.count=Gemdos(&H3F,S.handle, =>
L:Buff.size,L:Varptr(Buff.$))
      W.count=Gemdos(&H40,D.handle, =>
L:R.count,L:Varptr(Buff.$))
    Until R.count<Buff.size
    Rem CLOSE SOURCE+DESTINATION-FILE
    If Gemdos(&H3E,D.handle)<0
      Print "CAN'T CLOSE DESTINATIO =>
N-FILE"
    Endif
    Label:
    If Gemdos(&H3E,S.handle)<0
      Print "CAN'T CLOSE SOURCE-FILE"
    Endif
  Endif
  Eop.copy:
 Next I%
 Eop.copy_executer:
 C.continue=1
 @Load_cock
Return
' ************************************
Procedure Copy_errors
  If Err=-36
    Resume Next
  Endif
Return
' ************************************
Procedure E.xec
  @Save_cock
  @Executer
  If R.estore=1
    Reserve S.pace
  Endif
  @Load_cock
Return
' ************************************
Procedure Executer
  Fileselect "D:\*.*","",Selected$
  If Selected$>"" And Selected$<>"\"
    If Exist(Selected$)<>0
      S.pace=Fre(0)
      Reserve 50000
      Rem SOMMIGE PROGRAMMA'S  "MAKEN =>
" ZELF GEEN CURSOR,HEEFT HET
      Rem UIT TE VOEREN PROGRAMMA EEN =>
 CURSOR NODIG EN HET PROGRAMMA
      Rem MAAKT DEZE ZELF NIET, VOEG =>
DAN DE ONDERSTAANDE "ESCAPE CODE"TUSSEN.
      Rem     Print Chr$(27);"e"
      Exec 0,Selected$,"",""
      R.estore=1
    Endif
  Endif
Return
' ************************************
Procedure Delete
  Delete%=1
  @Copy
  Delete%=0
Return
' ************************************
Procedure Delete_executer
  For I%=0 To File_aantal%
    If Filenr.(I%)>0
      Source$=Mid$(D.isk$,1,3)+Mid$(F =>
ile$(I%),2,13)
      If Folder%=1
        Source$=Mid$(D.isk$,1,2)+Fold =>
er$+"\"+Mid$(File$(I%),2,13)+Chr$(0)
      Endif
      Kill Source$
    Endif
  Next I%
  Delete%=0
Return
' ************************************
Procedure Format
  @Save_cock
  Fout_format=0
  Buff$=String$(8*1024,0)
  F1$="|Indien U op OK klickt,worden"
  F1$=F1$+"|alle files op deze diskette"
  F1$=F1$+"|vernietigd"
  F2$="|DISKETTE TYPE|             "
  F3$="|DOE DE TE FORMATEREN DISKETTE"
  F3$=F3$+"|      IN Drive A"
  Alert 3,F1$,2,"EXIT|OK",F.ormat
  If F.ormat=1
    Goto Eop.format
  Endif
  Alert 1,F2$,2,"SINGLE|DOUBLE",N.side
  Alert 3,F3$,1,"OK|EXIT",F.ormat
  If F.ormat=2
    Goto Eop.format
  Endif
  For Side=0 To N.side-1
    Print At(5,8);"FORMATTING SIDE";' =>
'Side
    Rem FORMAT TRACK 0-79
    For Tr.nr=0 To 79
      Dymmy%=Fre(0)  !GARBAGE COLLECTI =>
ON VERPLICHT
      Print At(5,10);"Formatting Trac =>
k";''Tr.nr
      If Tr.nr<2 And Side=0
        Rslt=Xbios(10,L:Varptr(Buff$) =>
,L:0,0,9,Tr.nr,Side,1,L:&H87654321,&H0)
      Else
        Rslt=Xbios(10,L:Varptr(Buff$) =>
,L:0,0,9,Tr.nr,Side,1,L:&H87654321,&H =>
E5E5)
      Endif
      If Rslt<>0
        Print At(1,1);"ER IS IETS VER =>
KEERD GEGAAN BIJ TRACKNR.";Tr.nr
        Alert 1,"",1,"OK",Dummy
        Tr.nr=79
        Side=1
        Fout_format=1
      Endif
    Next Tr.nr
  Next Side
  If Fout_format=1
    Goto Eop.format
  Endif
  Rem CREATE BOOT_SEKTOR
  Print
  Print "CREATING BOOT_SEKTOR"
  Pause 30
  Buff$=String$(512,0)
  Poke Varptr(Buff$)+12,2
  Poke Varptr(Buff$)+13,2
  Poke Varptr(Buff$)+14,1
  Poke Varptr(Buff$)+16,2
  Poke Varptr(Buff$)+17,&H70
  If N.side=1
    Poke Varptr(Buff$)+19,&HD0
    Poke Varptr(Buff$)+20,2
    Poke Varptr(Buff$)+21,&HF8
  Else
    Poke Varptr(Buff$)+19,&HA0
    Poke Varptr(Buff$)+20,5
    Poke Varptr(Buff$)+21,&HF9
  Endif
  Poke Varptr(Buff$)+22,5
  Poke Varptr(Buff$)+24,9
  Poke Varptr(Buff$)+26,N.side
  Rslt=Xbios(18,L:Varptr(Buff$),L:&H1 =>
000000,N.side+1,0)
  Rslt=Xbios(9,L:Varptr(Buff$),L:0,0, =>
1,0,0,1)
  If Rslt<>0
    Alert 1,"| |SOMETHING WENT WRONG" =>
,1,"OK",Dummy
  Endif
  Eop.format:
  @Load_cock
Return
' ************************************
Procedure New_folder
  @Save_cock
  On Error Gosub Folder_err
  Print At(5,10);"NAAM FOLDER"
  Print At(17,10);
  Input "",Folder$
  Mkdir Folder$
  @Load_cock
Return
' ************************************
Procedure Folder_err
  If Err=-36
    Cls
    Alert 1,"|FOLDER BESTAAT REEDS",1 =>
,"OK",Dummy
  Endif
  Resume Next
Return
' ************************************
Procedure Compile
  @Save_cock
  S.pace=Fre(0)
  Reserve 50000
  If Exist("GFA_BCOMP.PRG")=0
    Alert 3,"| .PRG_FILE NIET OP|DE A =>
KTUELE DRIVE AANWEZIG",1,"OK",Dummy
    Goto Eop.compile
  Endif
  If Exist("GFA_BCOMP.RSC")=0
    Alert 3,".RSC_FILE NIET OP|DE AKT =>
UELE DRIVE AANWEZIG",1,"OK",Dummy
    Goto Eop.compile
  Endif
  Exec 0,"GFA_BCOMP.PRG","",""
  Eop.compile:
  Reserve S.pace
  @Load_cock
Return
' ************************************
Procedure Memory
  Mem%=Fre(0)
  B.oxtext$="|"+Str$(Mem%)+Space$(2)+ =>
"bytes"+"|"+Space$(1)+"|"+Space$(2)+" =>
free"
  Alert 1,B.oxtext$,1,"OK",Dummy
Return
' ************************************
Procedure D.free
  Dum$=Space$(15)
  Alert 1,Space$(15),3,"A|B|D",Free
  Free$=Str$(Dfree(Free))
  If Free=3
    Free$=Str$(Dfree(Free+1))
  Endif
  B.oxtext$="|DEZE DISK HEEFT NOG|"
  B.oxtext$=B.oxtext$+Free$+" BYTES|B =>
ESCHIKBAAR"
  Alert 1,B.oxtext$,1,"OK",Dummy
Return
' ************************************
Procedure Set_keyboard
  If Y=14
    Tel.=Rep%
  Else
    Tel.=Tak%
  Endif
  @Dec_inc
  If Tel.>16
    Tel.=0
  Endif
  If Tel.<0
    Tel.=16
  Endif
  If Y=14
    Rep%=Tel.
  Else
    Tak%=Tel.
  Endif
  Key$=Space$(2)
  Lset Key$=Str$(Tel.)
  Print At(15,Y);Key$
  Key.b%=Xbios(35,Rep%,Tak%)
Return
' ************************************
Procedure Key.board_init
  Dummy%=Fre(0)
  Key.b%=Xbios(35,-1,-1)
  Rep%=Peek(Varptr(Key.b%)+2)
  Tak%=Peek(Varptr(Key.b%)+3)
Return
' ************************************
Procedure Set_resolution
  If @Present_resolution=0      !laag
    Resolutie%=@Med_resolution
    Res_flag=1
  Endif
Return
' ************************************
Procedure Reset_resolution
  If Res_flag=1
    Resolutie%=@Low_resolution
    Res_flag=0
  Endif
Return
' ************************************
Procedure Delete_folder
  On Error Gosub Delete.folder_errors
  @Save_cock
  Cls
  Print At(5,5);"syntax: drive:\folde =>
r_naam"
  Print At(5,10);"NAAM FOLDER"
  Print At(17,10);
  Input "",Folder$
  Rmdir Folder$
  @Load_cock
Return
' ************************************
Procedure Delete.folder_errors
  If Err=102
    Resume Next
  Endif
  If Err=-36
    Alert 3,"|DE FOLDER MOET EERST|LE =>
EG ZIJN",1,"OK",Dummy%
    Resume Next
  Endif
Return
' ************************************ =>
****
Procedure More_errors
  If Err=26
    Resume Next
  Endif
Return
' end end end end end end end end end =>
 end end end end end end end end end
```

# MC68881 FLOATING POINT COPROCESSOR

***De firma Weide Electronics uit West-Duitsland brengt sinds enige tijd hardware uitbreidingen voor de Atari op de markt die op eenvoudige wijze (zonder solderen!) kunnen worden gemonteerd. In eerste instantie ging het daarbij om printjes waarmee ST's van een halve megabyte om te zetten waren in machines van een volle megabyte. Inmiddels bestaat ook de uitbreiding tot 2,5 megabyte, al is die nog redelijk prijzig. Maar wat wil je ook als er zestien chips van pakweg f 70,– inzitten. Voor ongeveer f 1.300,– is het geheel van u.***
***De nieuwste telg van Weide Electronivcs is een klein bordje met de MC68881 floating point coprocessor om rekenen sneller te maken. Deze chip is eigenlijk ontworpen voor de 68020 maar kan, zij het iets minder efficiënt, ook met de 68000 babbelen. Jos Vermaseren legde hem op de testbank.***

## BESTE BIBLIOTHEEK

Dank zij de duidelijke handleiding die werd bijgeleverd, was de MC68881 eenvoudig te installeren: twee van de roms eruit, clipje van de glue-chip, bordje bovenop de glue-chip, bordje in de voetjes van de roms, de roms daar weer bovenop, Atari weer in elkaar en klaar is kees.
Onmiddellijk daarna probeerde ik het bijgeleverde demonstratieprogramma. Dit geeft een vergelijking van de rekensnelheid met en zonder de coprocessor. De software floating point routines in de benchmarks zijn uit de bibliotheek van Megamax. Aangezien dit niet zo'n beste bibliotheek is (hetgeen Megamax zelf toegeeft, ik zeg hier niets nieuws), zijn de bijgeleverde benchmark resultaten wat overtrokken, tenzij je natuurlijk met Megamax werkt.
In het algemeen geldt voor de verhoudingen tussen de 68000 en 68881 dat snelheidsverbeteringen van meer dan een faktor honderd voor wiskundige funkties zoals log, exp, sin en dergelijke normaal zijn. Hierbij citeer ik de cijfers die Motorola zelf opgeeft. Bij normale operaties als optellen en vermenigvuldigen zijn de winsten minder spektakulair: er gaat veel tijd verloren aan het oversturen van de getallen van en naar de coprocessor. Een goede programmeur kan zelf optelroutines schrijven die evenveel tijd vergen als optellen met de coprocessor. Bij vermenigvuldigen wint de coprocessor met een faktor van meer dan twee en bij delen is de faktor drie, mits u een heel goede bibliotheek heeft. Ik heb vergeleken met de snelste bibliotheek die momenteel voorhanden is: de nieuwste update van Lattice.
Aangezien het oversturen zoveel tijd vergt, is de funktie sqrt (wortel) zelfs sneller dan vermenigvuldigen of optellen. Hier hoeft namelijk maar één argument naar de chip gestuurd te worden!

## ALTIJD RESULTAAT

Na deze standaardtests wilde ik de benodigde instrukties opnemen in mijn eigen versie van de Absoft Fortran bibliotheek. In principe is dit heel eenvoudig, maar al mijn rekenprogramma's overleden plots aan fouten.
En toen begon de ellende. Nadere bestudering van het testprogramma van Weide maakte onmiddellijk duidelijk dat de zaak werd opgelicht. Weliswaar stuurt het programma zowel getallen als de juiste instrukties naar de chip, maar vervolgens wordt er absoluut niet getest of de chip wel gedaan heeft wat ervan verlangd werd.
Foutmeldingen bij delen door nul negeert het programma bijvoorbeeld volledig: als de chip niet meer bezig is, wordt er aangenomen dat het resultaat staat te wachten. Dit wordt vervolgens zonder enige vorm van kontrole opgehaald.
Bij mij seinde de chip kontinu dat hij op tilt stond. Dat was dus niet het signaal dat hij bezig was en dus werd er aangenomen dat er een getal kon worden opgehaald. Zo ben je snel klaar met je berekeningen, dat wel. Kortom, niet een programma waar je je administratie of je bouwkundige berekeningen aan zou willen toevertrouwen.
De tests van Weide zijn blijkbaar alleen maar afgestemd op hoeveel aanroepen de coprocessor per sekonde aankan en niet wat die getallen betekenen en of alles wel goed gegaan is.
De benchmarks van Weide Electronics zijn dus onbetrouwbaar. Bovendien zijn de bijgeleverde bibliotheken niet van goede kwaliteit. Als iemand bij mij zoiets zou leveren zou ik hem ontslaan. De kwaliteit van de software is echter ondergeschikt, want er zijn genoeg frutselaars die in no time iets beters kunnen leveren.

## OP TESTTOERNEE

Aangezien ik de kode van Weide Electronics had gekopieerd om snel aan de slag te kunnen, stond ik opeens weer met lege handen. Er zat niets anders op dan om, met het boek in de hand, zelf een routine te schrijven. Toen bleek al snel dat de chip of zijn verbinding niet al te goed funktioneerde. Er kwamen allerlei willekeurige fouten. Soms ging dezelfde operatie wel goed, en soms niet. Chip eruit, goed kijken, chip terug en vervolgens konstateren dat het nu nog vaker fout ging. Op zo'n moment kun je het natuurlijk opgeven, maar aangezien dit het enige exemplaar in Nederland was (op testtoernee), ga je door tot vier uur 's nachts.
De routines werden bijgespijkerd zodat ze, als er een fout geconstateerd werd, de hele operatie weer herhaalden. Nu draaiden mijn programma's wel, maar de tests waren er een flink stuk trager op geworden (ongeveer een faktor twee). Voor de vermenigvuldiging betekende dit nog steeds een snelheidswinst en voor de deling nog steeds een grote winst.
De volgende dag de chip er weer uitgehaald, vuil van de pootjes verwijderd en het ding er weer in. Nu was de zaak er nog erger op geworden. Voor mij leek hiermee de zaak wel af, want dit betekent in mijn ogen dat de konstruktie om de MC68881 te monteren absoluut onvoldoende is.
U kunt wellicht mijn verbazing begrijpen toen een uurtje later hetzelfde testprogramma opeens een stuk sneller liep. Er traden echter nog wel degelijk fouten op. Nog wat wachten bleek niet nog meer te helpen. Blijkbaar vond de chip dat-ie nu wel warm genoeg was.
Ik heb toen in dezelfde stijl als de vermenigvuldig-, deel- en optelroutines er ook maar eens wat funkties in gestopt (met allerlei tests en instrukties als 'opnieuw beginnen bij onraad'. Dat was smullen. In de tabel vindt u een 'benchmark' resultaat voor de zogenaamde Savage benchmark, die onlangs erg populair was.

## TABEL

De resultaten zijn (alles in dubbele precisie):

| | | |
|---|---|---|
| Oude Lattice | > 200 | sekonden |
| Mark Williams | 83 | sekonden |
| Absoft | 68 | sekonden |
| Pro Fortran | +/- 50 | sekonden |
| Nieuwe Lattice | 39.8 | sekonden |
| Haalbaar moet zijn | +/- 25 | sekonden |
| Absoft met coprocessor | 1.74 | sekonden |

En dan liep de coprocessor dus nog enigszins mank. De meeste tijd gaat nu verloren doordat alle tussenresultaten iedere keer teruggehaald en dan weer opgestuurd worden. Als je er even in assembler aan gaat staan, kan de hele operatie in minder dan een sekonde! Alles vindt dan in de registers van de coprocessor plaats. Deze heeft voor het rekenen namelijk maar 0.36 sekonde nodig plus de tijd om de instrukties te ontvangen. Dat laatste gaat veel sneller dan het sturen van getallen van acht bytes in de vorm van twee pakketten. Ik had helaas geen tijd om deze test ook nog uit te voeren. Ik schat dat het dan in minder dan 0.7 sekonde gaat.

Ik neem aan dat er in de toekomst wel een betrouwbaardere versie van de coprocessor komt. Dan wordt het heel leuk om ermee te werken. Typische berekeningen die in dubbele precisie floating point werken en alleen 'basis operaties' gebruiken, worden meer dan twee maal zo snel. Als uw bibliotheek niet zo goed is, kan de winst zelfs oplopen tot een faktor tien.
Ik heb de 68851 's nachts zo'n rekenjob laten uitvoeren en in plaats van tien uur (twee nachten eerder) deed het programma er nu nog vijf uur over. Hierbij dient te worden opgemerkt dat het bij een 'goede' coprocessor waarschijnlijk nog sneller gaat. Als er heel veel gebruik wordt gemaakt van de ingebouwde funkties, zoals bijvoorbeeld bij grafische pakketten waarbij even alles geroteerd moet worden, dan is de winst meer te vergelijken met die in de Savage test. Nog beter wordt het voor operaties die veel rekenwerk doen met weinig getallen. Dan loont het immers de moeite om een speciale routine te schrijven die alles in de registers van de coprocessor kan uitvoeren.

*Mijn konklusie: het is een genot om met de MC68881 te werken, maar de bijgeleverde software en de konstruktie laat nog te wensen over.*
*De Weide produkten worden in Nederland ingevoerd door Commedia.*




***Een hifi-installatie, een videocamera en een scanner zijn drie randapparaten die zich gemakkelijk met de ST laten verbinden. De software die dit mogelijk maakt, gaat altijd vergezeld van wat hardware (de interface). START bekeek de ART-TIZER, de PRO SOUND DESIGNER en MRT. Digitaliseren op elk gebied. Een verkenning.***

## 1 PRO SOUND DESIGNER

### BELOFTE

De PRO SOUND DESIGNER is een produkt van de firma Eidersoft en werkt voorlopig alleen in lage resolutie. De interface, het hardware gedeelte van het pakket, is met haast op de markt gebracht. Het printplaatje wordt nog niet door een kastje beschermd. Eidersoft belooft geregistreerde eigenaren gratis zowel een kastje als een hoge resolutie versie van de software toe te zenden.




De interface moet aangesloten worden op de printerpoort. Het vervelende van de printerpoort is dat de ontwerper van de interface er geen voeding vandaan kan halen. Men heeft er daarom voor gekozen om met een batterijtje van 9 volt te werken. Wie een losse transformator bezit, kan die natuurlijk ook gebruiken. De vraag is alleen of de ontwerper ons al deze toestanden niet had kunnen besparen. Via een plug in de joystickpoort is bijvoorbeeld in ieder geval 5 volt af te tappen. Maar misschien is voor de aparte voeding gekozen om de ST en de interface technisch zoveel mogelijk gescheiden te houden.

De Pro Sound Designer kan via een standaard koptelefoonplug met de uitgang van een versterker verbonden worden. Afspelen geschiedt via de luidspreker van uw monitor. Er is echter een extra chip te koop die afspelen via uw stereo mogelijk maakt. Het printplaatje bevat al een plaats waar die chip eenvoudig ingestoken kan worden.

### MEL EN KIM

De software van de Sound Designer bestaat eigenlijk uit twee delen: een sampler en een geluidschip editor. Allereerst de sampler. Hiermee is het mogelijk een stuk muziek of een stem met de ST op te nemen. De software digitaliseert het inkomende geluid. Hoe vaker de ST dat per sekonde doet, hoe beter de kwaliteit. Het nadeel van 'vaak' is echter dat het geheugen van de ST snel vol is. De basissnelheid van de sampler is goed voor ongeveer dertig sekonden muziek op een gewone 520 ST. Voor een kompleet singletje zullen we dus op een Mega ST moeten wachten.
Voor de test verbonden we de interface met een compact disc speler. Een stuk "Dire Straits" werd moeiteloos en met een zeer acceptabele kwaliteit gedigitaliseerd. Het minuutje geluid dat in mijn ST(+) paste, legde beslag op maar liefst 384.000 bytes. De aanschaf van een extra doosje diskettes is dus aan te bevelen.
Als het geluid eenmaal 'gevangen' is, kan men het beperkt manipuleren. Twee 'markers' maken het mogelijk een stuk geluid af te bakenen. Zo'n 'blok' kan de gebruiker dan aan een funktietoets toewijzen.
Elk moment dat op zo'n funktietoets gedrukt wordt, herhaalt de ST het gemarkeerde stuk. Nogmaals drukken tijdens het spelen start het fragment opnieuw. Dit geeft mogelijkheden tot beperkte 'Mel en Kim'-effekten ("T-T-T-Take or leave me..")

### FAST BASIC

Aardiger is de kopieerfunktie. Een blok geluid kan zowel vervangend als transparant gekopieerd worden. Vervangend wil zeggen dat de plaats die het blok toegewezen krijgt, voor het kopiëren gewist wordt. Een transparante kopie laat de oorspronkelijke informatie intakt. Dit maakt nagalmeffekten mogelijk.
Daarmee hebben we het manipuleren van het geluid eigenlijk gehad. Wat rest is een aantal grapjes. Zo kan een mikrofoon zo worden afgesteld, dat bij het bereiken van een bepaald geluidniveau het gedigitaliseerde geluid afgespeeld wordt. Een manier om inbrekers af te schrikken. Of ze op het bestaan van uw kostbare apparatuur te wijzen natuurlijk.
Op dezelfde manier kan de software bij overschrijding van een bepaald geluidsniveau gaan opnemen.
Achteruit afspelen is ook mogelijk. Daar is de pret echter snel vanaf.

Als tweede funktie geeft de Pro Sound Designer de mogelijkheid de geluidschip van de ST rechtstreeks te programmeren. Via het instellen van de drie ter beschikking staande kanalen, waveform, sustain, release, enz. kan een geluidseffekt gekreëerd worden dat de gebruiker in een eigen programma kan gebruiken.
De uitvoer van het programma is naar keuze C, machinetaal of Fast Basic. Nog geen GfA. Dat moet in Engeland de macht nog grijpen.
De konklusie mag zijn dat de Pro Sound Designer zijn werk goed doet, maar dat de mogelijkheden om met het gedigitaliseerde geluid te experimenteren wat beperkt zijn.

## 2 ART-TIZER

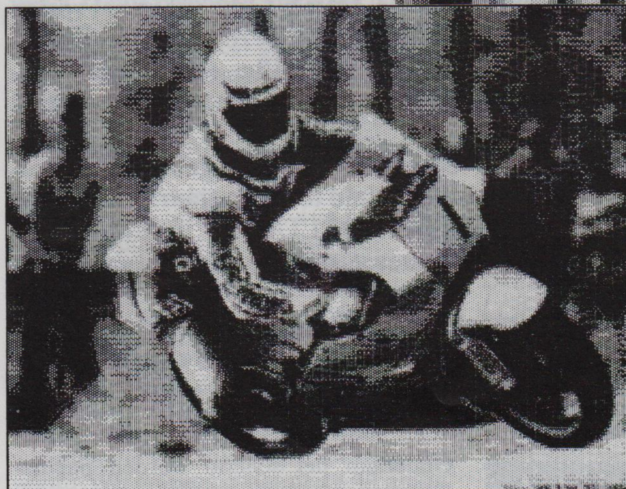
### KLEUREN MET RASTERS

Het digitaliseren van een videosignaal is de taak die de Art-Tizer zich gesteld heeft. De interface, een produkt van de Duitse firma Print Technik, moet op de ROMpoort aangesloten worden. De hardware zit keurig in een kastje en laat zich door de ST zelf voeden.
De bijgeleverde software tovert het signaal van een videokamera gemakkelijk om in een plaatje. Dat kan zowel in hoge als in lage resolutie. Veel mogelijkheden heeft de software niet, maar de gebruiker kan de plaatjes gemakkelijk met een tekenpakket






verder bewerken. De beelden kunnen in het formaat van Neochrome, Degas, Doodle of Art-Director op diskette gezet worden.
In zwart/wit heeft de gebruiker de beschikking over een palet van zestien rasters. Het binnenkomende beeld wordt aan de hand van dit palet 'ingekleurd'. Wie het pakket in lage resolutie gebruikt, heeft uiteraard echte kleuren tot zijn beschikking. Daar het binnenkomende signaal alleen zwart/wit is, is dit alleen nuttig voor speciale effekten. De standaardinstelling is dan ook terecht een zwart/wit palet.
Afbeeldingen in lage resolutie zijn van hogere kwaliteit dan die in hoge resolutie. Ik denk dat dat komt omdat de kleurenmonitor zestien grijswaarden gebruikt, terwijl de software in hoge resolutie de standaard GEM-rasters gebruikt. En dan moet je wel een erg pokdalig huid hebben voor je van een natuurgetrouwe afbeelding van een gezicht kunt spreken.
Al met al vielen de resultaten me nogal tegen. De gedigitaliseerde afbeelding van de twee STARTS die u hierbij aantreft, laat weinig van onze mooie covers over. Eerlijkheidshalve moet ik zeggen dat de opname met een heel eenvoudige kamera gemaakt is. Maar daar de Art-Tizer gezien zijn prijsstelling zich juist tot het low budget gedeelte van de markt richt, zullen er meer mensen een dergelijke kombinatie overwegen.



## 3 MRT

### PROEFKONIJN

Van MRT werden we in de gelegenheid gesteld een voorversie te bekijken. Dat is niet zo verwonderlijk, want Egon Lubbers, één van de auteurs, is een trouwe START-medewerker. Samen met Pim Borst, die de hardware voor zijn rekening nam, werd een kompleet systeem ontworpen om tekeningen of foto's te scannen.
Die foto's moeten puntje voor puntje door een printer afgetast worden. MRT verandert dus een uitvoerorgaan in een invoerorgaan. Daarvoor moet de gebruiker op de kop van de printer een lichtgevoelige cel bevestigen. Voorlopig gaat dat met plakband, maar wellicht wordt er op het moment dat het produkt op de markt komt een klemmetje bijgeleverd.
Aan de kant van de ST moet zowel een verbinding met de RS232-, als met de joystickpoort gemaakt worden. Dit laatste is alleen nodig om de interface te voeden. Dan is het tijd om de software op de printer af te regelen. Het tempo waarmee de printer over de tekening jaagt en de wachttijd aan het begin van elke nieuwe regel moeten zo nauwkeurig mogelijk bepaald worden. De voorlopige handleiding gaf weinig handvaten voor mijn Epson FX-80, zodat het enkele uren experimenteren kostte voordat de waarde van de parameters ongeveer bepaald was. Afbeeldingen in spiegelbeeld, langgerekte afbeeldingen, raadselachtige vlekken: er is heel wat ellende langsgekomen.
Te zijner tijd zal de consument het gemakkelijker krijgen. De bloedig verkregen cijfers van mijn FX komen in de definitieve handleiding en zo zijn er meer proefkonijnen.
Het resultaat van MRT mag er zijn. Ook hier treffen we mooiere plaatjes bij gebruik in lage resolutie, maar het verschil is minder groot dan bij de Art-Tizer.

### LOCAL FILL

De software van Egon is bijzonder verzorgd. Zo biedt hij maar liefst drie manieren aan om te digitaliseren. De eerste manier maakt gebruik van een aantal zelf in te stellen grijswaarden. De tweede methode werkt ook zo, maar mengt eventueel zelf twee grijswaarden door elkaar als de software op een bepaald punt geen bevredigende oplossing in het palet vindt.
De laatste manier digitaliseert in blokken. Wie de film "Westworld" gezien heeft, weet wat ik bedoel. Anderen moeten maar even naar de illustratie kijken. De grootte van de blokjes is vrij in te stellen.
Als het beeld eenmaal gescand is, kan er nog van alles aan veranderd worden. De mooiste optie van MRT is wel de mogelijkheid om alsnog een aantal kleuren uit het palet door één kleur te vervangen. Dat is bijvoorbeeld handig wanneer een achtergrond niet helemaal egaal is. Zoek in het palet de boosdoeners op, kies de kleur of grijswaarde die gebruikt moet worden en trek een rechthoek over de besmette plek op het beeld. Overal in de rechthoek waar de 'vlekkerige' kleuren voorkomen, worden ze vervangen. De rest van de rechthoek (en alles buiten de rechthoek) blijft intakt. Zo kan dus een kleur die op de ene plaats perfekt is en op een andere plaats stoort, lokaal gewijzigd worden.
De afbeeldingen kunnen op schijf gezet worden in de formaten van Art Director, Degas, Degas Elite (samengeperst), Neochrome en Doodle. Bovendien krijgt de koper een konversieprogramma ter beschikking om de zogenaamde 'scandata' geschikt te maken voor manipulatie door AIM (een ijzersterk public domain beeldbewerkingsprogramma).

*De Pro Sound Digitizer en de Art-Tizer worden gedistribueerd door PDS. Waarschijnlijk zal ook de definitieve versie van MRT door deze firma in omloop gebracht worden.*

*Jan van Die*

# DE BETERE TEKSTVERWERKERS

**De nieuwe generatie tekstverwerkers voor de Atari ST, die wij in het vorige nummer van START aankondigden, hadden we op deze plaats graag uitgebreid getest. Tot nu toe is echter alleen 1st Word Plus officieel verschenen. Microsoft Write, de ST-versie van Word, laat nog op zich wachten.**
**Beckertext ST zal voorlopig niet in Nederland worden uitgebracht. De release van WordPerfect ST tenslotte, is uitgesteld tot het najaar. Maar door bemiddeling van de Nederlandse uitgever, DELTAware in Rotterdam, kreeg START de beschikking over de meest recente beta-versie uit de VS.**
**Wat een vergelijkend warenonderzoek tussen De Betere Tekstverwerkers had moeten worden, blijft in dit nummer beperkt tot een evaluatie van 1st Word Plus (2.02) en een voorbespreking van WordPerfect.**

## WordPerfect op de ST: the best of two worlds

De aankondiging dat er een muisgestuurde GEM-versie voor de Atari ST komt van het bekende tekstverwerkingspakket WordPerfect, betekent meer dan zomaar de zoveelste annonce. WordPerfect Corporation is een van de meest toonaangevende softwarenhuizen van dit moment. Het feit dat dit bedrijf de ST gaat ondersteunen, kan - juist op het moment dat Atari de Mega ST's introduceert - een flinke duw in de richting van de zakelijke markt betekenen. En dat is precies waar Atari met de Mega's naar toe wil.
WordPerfect (WP) wordt op de MS-DOS computers (IBM-PC's en compatibelen) zo'n beetje beschouwd als het neusje van de zalm op het gebied van tekstverwerking. Het pakket staat bovendien bekend als gebruikersvriendelijk.
Wie een ST of een Apple Macintosh gewend is, zal bij die kwalifikatie de schouders ophalen, want 'gebruikersvriendelijk' op een PC betekent altijd nog: werken met de handleiding naast het toetsenbord om allerlei kodes en toetsenkombinaties op te zoeken. De vraag is dan ook of het huwelijk tussen de redelijk gemakkelijk te hanteren ST en WordPerfect met z'n zeer uitgebreide mogelijkheden, vruchtbaar zal blijken. Want als dat het geval is, zou WordPerfect ST werkelijk 'the best of two worlds' zijn.
Mijn eerste indruk is dat de makers van WP er zeer behoorlijk in geslaagd zijn dit ideaalbeeld te benaderen. Uit alles blijkt dat het pakket bedoeld is voor mensen die veel en verschillende teksten moeten verwerken. Een van de belangrijkste voordelen die WP naar mijn smaak heeft ten opzichte van bijvoorbeeld 1st Word Plus is het feit dat het leeuwedeel van de funkties zowel door middel van een toetsaanslag als met behulp van de muis kan worden aangeroepen. Geroutineerde tikkers halen hun vingers liever zo min mogelijk van het toetsenbord af en gebruiken een muis alleen om tekstblokken en dergelijke te definiëren.

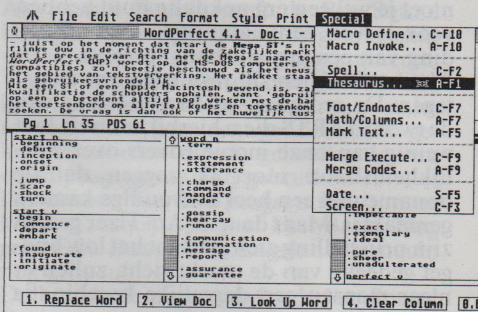
### BLOKKEN TERUGROEPEN

Het startscherm ziet eruit zoals dat van de meeste GEM-applikaties: een simpele menubalk waarin de opties vermeld staan. Onderaan het scherm worden paginanummer en cursorpositie aangegeven. Aan dit laatste cijfer is tevens te zien welk lettertype wordt gebruikt.
Het valt meteen op dat WP geen gebruik maakt van het standaard systeem-font van de ST (8x16 pixels), maar van het middelgrote font (8x8 pixels). Dat heeft als voordeel dat er meer karakters op het scherm afgebeeld kunnen worden, maar het komt de leesbaarheid bepaald niet ten goede: de letters zijn nogal aan de kleine en vette kant.
De schermuitvoer is volledig 'WYSIWYG' (What You See Is What You Get): vette, kursieve en onderstreepte letters zijn als zodanig zichtbaar, net zoals bij 1st Word Plus. Ook het werken in kolommen, maximaal vijf naast elkaar, is geïmplementeerd. WP kan vier dokumenten tegelijkertijd in het geheugen houden, waartussen uitwisseling van tekstblokken mogelijk is. Bijzondere lettertekens en symbolen die niet op het toetsenbord staan, kunnen via een speciaal menu onder een kombinatie met de Alternate- of Controltoets worden gezet.

Een stuk tekst per ongeluk weggooien is bij WP geen onherstelbare ramp. Door middel van de optie 'undelete' kunnen de laatste drie tekstblokken die gewist zijn, teruggeroepen worden. Ook op andere gebieden ondervangt het programma veel risiko's. Om een (instelbaar) aantal sekonden worden de wijzigingen die in een dokument zijn aangebracht, automatisch op schijf gezet. En raakt het geheugen van de computer onverhoopt vol, dan maakt WP zelf ruimte door een deel van de tekst tijdelijk op een (ram)disk te zetten.

### SYNONIEMENBOEK

WordPerfect is een zeer omvangrijk pakket, dat vrijwel alles in zich heeft wat op het gebied van tekstverwerking is bedacht. Om een indruk te geven: de beta-versie die wij ontvingen beslaat drie propvolle dubbelzijdige diskettes; in totaal meer dan anderhalf Megabyte programmakode en hulpbestanden. Daarbij inbegrepen zeer uitgebreide drivers voor 373 lettertypen en 264 printers, kompleet met breedtetabellen voor proportionele fonts.
Alleen al het - voorlopig alleen nog Engelse - woordenboek omvat maar liefst 115-duizend woorden (300 K). En dan is er nog de ruim 360 K grote 'Thesaurus': een soort on-line synoniemenboek dat bij een opgegeven woord in luttele sekonden alle synoniemen, bijna-synoniemen, verwante woorden en antoniemen (tegenovergestelden) zoekt. Voor journalisten, tekstschrijvers e.d. een fantastisch hulpmiddel.
Professionele gebruikers zullen het waarderen dat de ST dokumenten die met de PC-versie van WordPerfect zijn gemaakt, kan verwerken zonder dat daarbij de tekstformattering verloren gaat. Ook andersom kunnen teksten moeiteloos overgezet worden; helemaal nu ook IBM, net als Atari, heeft gekozen voor het 3,5 inch disketteformaat.




### MENU ACHTER MENU

De keuze- en instelmogelijkheden zijn zo uitgebreid, dat het al snel gaat duizelen. Achter vrijwel elk menu van WP bevindt zich weer een nieuw keuzemenu, meer dan eens gevolgd door weer een menu. Maar waarschijnlijk zal slechts een enkeling al die verschillende opties ook daadwerkelijk gebruiken.

De meest opmerkelijke kwaliteiten van WordPerfect vinden we onder het Special-menu. Hier schuilt de werkelijke kracht van het pakket. Behalve de al genoemde Thesaurus kunnen via dit menu opties worden geaktiveerd als spellingkontrole, makro's (een reeks opdrachten die onder een toets gezet wordt), voet- en eindnoten, berekeningen binnen de tekst, hoofdstuken paragraafindeling, inhoudsopgave en trefwoordenregister en koppeling van tekst aan gegevens uit een database ('mailmerge').

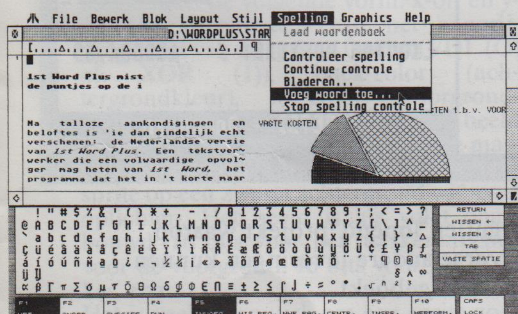
### KONKLUSIE

*Een definitief oordeel over WordPerfect ST is op dit moment niet mogelijk omdat het pakket zich nog in de fase van ontwikkeling bevindt. Maar zelfs de voorversie, die uiteraard nog verre van 'perfect' is, maakte al grote indruk door de ongekende mogelijkheden, de logische opbouw en het bedieningsgemak.*
*Het pakket wordt naar alle waarschijnlijkheid in het late najaar in de VS en kort daarna ook in Nederland uitgebracht. Het gaat dan overigens nog om een geheel Engelstalige versie. Of er een Nederlandstalig programma komt, zoals voor de PC het geval is, hangt af van de vraag. De verkoopprijs bedraagt bij introduktie f 950,- (exklusief BTW).*
*Informatie bij DELTAware BV, telefoon (010) 4 55 00 66.*

# 1st Word Plus mist de puntjes op de i

Naar de opvolger van 1st Word, de standaard-tekstverwerker voor de Atari ST, is lange tijd reikhalzend uitgekeken. Het programma heeft zijn schaduw vooruitgeworpen door een groot aantal voor versies, die via het kopieercircuit verspreid werden. Vrijwel iedere ST-gebruiker heeft vroeg of laat wel zo'n illegale 1.X-versie in z'n bezit (gehad). Veel verrassingen heeft de uiteindelijke Plus-versie 2.02 dus niet meer, behalve dat alles echt werkt.
Over de handleiding niets dan goeds. Die is uitstekend verzorgd, helder geschreven en laat weinig aan het toeval over. Behalve 1st Word Plus behandelt de handleiding ook het meegeleverde programma 1st Mail, waarmee bestanden uit een database kunnen worden gekoppeld aan een tekst die met 1st Word Plus is gemaakt.

De eerste indruk van het programma zelf is eveneens meer dan uitstekend. Dit is een tekstverwerker die gewoon doet wat je wilt, zonder al te veel nukken of onlogische handelingen. Ten opzichte van 1st Word heeft de Plus-versie een groot aantal verbeteringen en uitbreidingen. Over het geheel genomen geldt dat het stuk voor stuk werkelijke verbeteringen zijn.
De meest opvallende nieuwe opties zijn: mogelijkheid om tekeningen of grafieken in de tekst te plaatsen en mee te printen, half-automatisch afbreken van woorden, kontrole van spelling met Nederlands woordenboek, voetnoten onderaan de pagina en gebruik van pagina-formats.




Voor het invoegen van grafische beelden wordt het programmaatje Snapshot als accessoire meegeleverd. Hiermee kan vanuit vrijwel elk GEM-programma een 'foto' van het scherm of een gedeelte daarvan worden gemaakt. Deze wordt op disk gezet en later op de juiste plek in de tekst gepositioneerd. Dit werkt uitermate eenvoudig en snel. Eenmaal in de tekst kan de grootte van het beeld niet meer aangepast worden (al is daar, via het tekenprogramma Degas Elite, wel een mouw aan te passen).
Het afbreken van woorden werkt eveneens gemakkelijk. Bij elk woord dat niet op een regel past, verschijnt een dialoogbox waarin het betreffende woord staat, voorzien van een afbreek-voorstel. In de meeste gevallen is die afbreking al goed. Zo niet, dan kan het koppelteken met een paar muisklikken naar de juiste plek gedirigeerd worden. Jammer alleen dat deze afbrekingen niet onmiddellijk in het woordenboek worden opgeslagen; nu moet je bepaalde woorden telkens opnieuw verbeteren.

## *WOORDENBOEK: ZWAKSTE SCHAKEL*

Over het woordenboek gesproken, dat is verreweg het zwakste onderdeel van 1st Word Plus, met name in de Nederlandse versie. Om te beginnen duurt het inladen van het woordenboek (dat meer dan 40.000 woorden omvat en 127 K groot is) ergerniswekkend lang. Van diskette doet het programma er maar liefst 88 sekonden over om de 127 K in te laden; zelfs bij gebruik van een harddisk is er nog 47 sekonde nodig.
Maar ook in andere opzichten schiet het woordenboek tekort. Om met het meest merkwaardige te beginnen: de mogelijkheid om woorden te schrappen ontbreekt domweg in de Nederlandse versie. Kennelijk gewoon vergeten, want in de handleiding wordt er wel degelijk gesproken over het kommando 'Verwijder woord'. Inplaats daarvan heeft 1st Word Plus bij de Nederlandse bewerking de optie 'raden' meegekregen. Deze optie zoekt woorden die veel lijken op het opgegeven woord. Wel grappig, maar niet echt nuttig. Bovendien ontbreekt tekst en uitleg hierover in de handleiding.
Maar het meest heb ik me geërgerd aan de inhoud van het woordenboek. Er zitten woorden tussen die absoluut onzinnig zijn in een spellingschecker, terwijl talloze woorden die iedereen regelmatig in teksten gebruikt, weggelaten zijn. Het is misschien flauw, maar het kostte me slechts luttele minuten om een kompleet abc te maken van veelgebruikte woorden die niet in de lijst voorkomen: anderen, beste, commercie, doen, eerdere, financiële, goede, herkent, industriële, ja, knallen, leren, meeste, nieuw, oplossing, prijs, quote, relaties, speciale, tip, u, vele, weten, xyleen, yen, zool.
En net zo'n lijst samenstellen met nietgangbare of zelf absolute onzinwoorden was nog veel makkelijker: aalbezie, baakgeld, cabolzuur, daaltijd, ebaucheren, faalfrequentie, gaaischieten, haagappel, ichtyol, jaaghout, kaaigeld, laarzeknecht, maagcatarre, naaktijlen, objectivite, paaltjeverwisselen, quadrupeden, raaigras, saffiaan, taalbeweging, uierboord, vaandrigsexamen, waarborde, xifold, yggdrasill, zwoerdsuperbase.
Meneer Van Dale zou zich in z'n graf omdraaien...
Nog een tekortkoming van de spellingschecker is het feit dat veel woorden niet helemaal juist gespeld zijn. Zo ontbreken regelmatig de trema's en andere accenten, waarmee 1st Word Plus verder juist zo elegant omspringt. De puntjes op de i ontbreken letterlijk en figuurlijk, en dat is ronduit slordig. Er moet ergens tussen Londen en Vianen iets grondig mis zijn gegaan.
Het taalgebruik is toch al niet de sterkste kant van de Nederlandse 1st Word Plus. Met name de schermmededelingen (alert- en dialoogboxen) vertonen soms nogal krom taalgebruik of zijn maar gedeeltelijk vertaald.
Een zwak punt van een geheel andere orde is de snelheid waarmee de tekst over het scherm scrolt, ook al is die ten opzichte van 1st Word wel enigszins verbeterd. Veel GEM-programma's lijden onder deze traagheid, maar dat het wel degelijk anders kan hebben inmiddels al verschillende andere tekstverwerkers (Textomat, BeckerText ST) bewezen. Absolute topsnelheid zag ik onlangs in de programma-editor Tempus; geen echte tekstverwerker, maar voor het korrigeren van eenvoudige teksten een lust om mee te werken!
Mijn laatste punt van kritiek: het programma is ten opzichte van zijn voorganger(s) wel erg kritisch geworden bij het laden van tekst. Meer dan eens is het mij overkomen dat 1st Word Plus een simpel stuk ASCII-tekst niet wilde slikken. Je krijgt dan de koele mededeling 'Geen geldige ASCII of GST tekstfile' en daarmee is het afgedaan. Overigens: ondanks de aankondiging dat 1st Word Plus teksten die met bepaalde beta-versies (1.X) zijn gemaakt niet zou willen laden, heb ik daar tot nog toe geen enkel probleem mee gehad.

## *BRIEVEN AAN DE LOPENDE BAND*

Het bijgeleverde 'mailmerge'-programma 1st Mail is wat minder soepel in de omgang dan 1st Word Plus. Zonder handleiding kom je sowieso nergens en ook met is het nog even doorbijten. Maar de mogelijkheden - het op verschillende manieren koppelen van een tekst aan een database-bestand - zijn er dan ook naar. Dit computerkunstje wordt het meest gebruikt in brieven aan bijvoorbeeld een groot aantal klanten, maar ook veel ingewikkelder en omvangrijker toepassingen zijn denkbaar.
Aan de hand van een aantal voorbeelden heb je toch binnen afzienbare tijd je eerste 'mailmerge' op poten. Eerst wordt in 1st Word Plus een moederdokument gemaakt. Op de daarvoor met aangegeven plaatsen vult 1st Mail vervolgens de gevraagde gegevens in. Het gemakkelijkst is natuurlijk om die uit een database te halen, maar daarvoor moet nog wel het een en ander gebeuren.
Helaas beschrijft de handleiding alleen hoe 1st Mail samenwerkt met H & D Base (in Nederland vrijwel niet gebruikt), LaserBase en DB Master One. Zelfs Atari's eigen database Adimens, dat op de verpakking als „zinvolle aanvulling" wordt aangeprezen, ontbreekt. Hetzelfde geldt voor in ons land populaire databases als DB Man en Astodat II.

## KONKLUSIE

*Ondanks een aantal slordigheden op met name het taalkundig vlak is 1st Word Plus een bijzonder plezierig en volledig tekstverwerkingsprogramma. Vrijwel alle denkbare mogelijkheden zijn aanwezig en gemakkelijk te bedienen.*
*Dat het woordenboek allesbehalve probleemloos is, mag geen reden zijn om 1st Word Plus naar de prullenbak te verwijzen. In het vorige nummer van START zijn al enkele uiterst zinvolle hulpprogramma's ten doop gehouden, waaronder een uitstekend werkende spellingschecker. 1st Word Plus heeft verder zoveel goeds in huis, dat het zijn ereplaats als standaard-tekstverwerker voor de Atari ST wat mij betreft mag houden. Met een verkoopprijs van f 239,- (inklusief BTW) biedt het heel veel waar voor z'n geld.*

● *Alfred Konijnenbelt*

# Projecta-4 / Faktura-4 is boekhouden met plezier.

Debiteuren
Krediteuren
Kas, bank, giro
Balans - V&W
Voorraad
Uren-registratie
Projekt-administratie
Fakturering

Projecta-4 / Faktura-4 is een van de plezierigste softwarepakketten op financieel gebied voor MS-DOS computers en de Atari-ST. Het heeft een ongekend groot gebruiksgemak en biedt standaard alles wat u van een serieus zakelijk pakket mag verwachten. Dat betekent dat u er voor nu en in de toekomst volledig mee uit de voeten kunt zonder eindeloos te moeten blijven investeren. Sterker nog: ons service-kontrakt geeft u het recht op gratis nieuwe versies als ons pakket straks nog uitgebreider wordt. En het geeft u tevens de zekerheid van een jaar lang uitstekende support! En dat alles voor een uiterst aangename prijs. Geen wonder dat er nu al honderden tevreden gebruikers zijn!

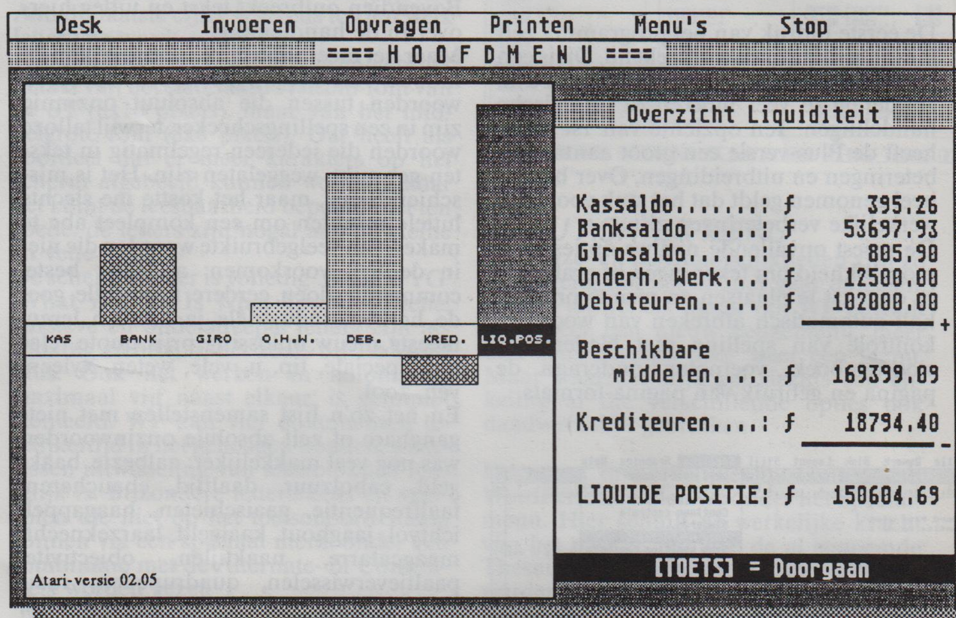
## UIT DE VAKPERS:

Het is natuurlijk bijzonder aardig om positieve reakties te krijgen op een produkt waarin je als maker gelooft. Zeker als de gebruikers - maar ook de vakmensen - onze filosofie achter het produkt blijken te herkennen.

Onder het kopje 'Projecta-4 - Management-gericht' schrijft P.C.M., het toonaangevende computerblad:
*"De handleiding van Projecta-4 wint met mijlen voorsprong de prijs voor het mooiste uiterlijk en ook ruimschoots die voor de best geschreven tekst (...)."*
P.C.M. mei 1987

Uit START (Atari-blad) mei 1987:
*"Het is een werkelijk een plezier om de dagelijkse boekingen in te voeren. (...) Fouten maken is nauwelijks mogelijk. (...)"*

Uit Atari-magazine, april 1987:
*"Deze Rolls Royce onder de boekhoudprogramma's is voor iemand die meer dan een kas-, giro-, en bankboek wil een MUST en zijn geld dubbel en dwars waard."*




## NIEUWS

Projecta-4 heeft nu standaard het decimale rekeningschema met zescijferige vrije grootboeknummers.

Nieuw is de mogelijkheid om in Faktura de faktuur te verlengen met een acceptgirokaart, waarop alle gewenste gegevens automatisch worden ingevuld.

Voor administratiekantoren is nu de licentie-versie beschikbaar waarmee een onbeperkt aantal boekhoudingen kan worden gevoerd.

Aan Faktura-4 is een extra optie toegevoegd van offertes en orderbevestigingen. Daarbij is het uiteraard mogelijk offertes direkt om te zetten in fakturen, maar indien nodig kunnen ze ook worden gewijzigd en aangevuld en zelfs worden omgezet in een regulier 'lopend' projekt waarop diverse andere leveringen kunnen worden geboekt.

In voorbereiding is een versie voor de gespecialiseerde boekhandel met een zeer uitgebreid ge-indexeerd artikelenbestand en zoekmogelijkheden op talloze velden.

Projecta-4 en Faktura-4 zijn nu ook verkrijgbaar in Belgie in een versie die volledig is aangepast aan de Belgische wettelijke normen.

Eerst zien en dan geloven? Stap binnen bij een van de ca. 120 dealers in Nederland en Belgie en vraag vrijblijvende demonstratie. Of bel TOPIC Software Amsterdam: 020-681400 voor een dealerlijst.

TOPIC Software - Ruyschstraat 23 - 1091 BR Amsterdam

(advertentie)

# DE SPRITES SPRUITEN NIET MEER!

**Het zit in uw ST, de meeste spelletjes hebben het, maar u heeft het nog nooit goed kunnen gebruiken. Altijd bleven er restjes achter of knipperde het beeld. Waar hebben we het over? Juist: de sprite!**
**Een sprite is een klein zelf gedefinieerd tekeningetje dat de programmeur over het beeld kan laten bewegen. In 'Sprites en Spruiten' (START 4) vertelde Peter Witvoet al het één en ander over de problemen die hij ondervond bij het programmeren van sprites in GfA-BASIC. Dit signalerende artikel krijgt nu een prettig vervolg: Egon Lubbers vond een manier om de mankementen te verhelpen.**

## EDITOR

Voor wie START 4 gemist heeft, volgt hier een korte samenvatting van de theorie.
Een sprite bestaat uit 16 bij 16 pixels. Spritedate heeft de volgende vorm: x-off en y-off (de koördinaten van het aangrijpingspunt van de sprite), format (VDI (O) of XOR (1)), back-color (achtergrondkleur), for-color (voorgrondkleur) en het eigenlijke beeld. Het beeld moet ook data voor het zogenaamde masker bevatten. Dit masker geeft aan waar de sprite op een zwarte achtergrond wit moet oplichten. Het beeld is als volgt opgebouwd: 16 bits voor het masker, 16 bits voor de voorgrond, 16 bits voor het masker, enz...
Het is veel rekenwerk om een sprite zonder hulpmiddelen te ontwerpen. Vandaar dat er in veel public domain bibliotheken een sprite editor aanwezig is. Met een sprite editor kunt u een sprite tekenen en deze in de vorm van kant en klare data op diskette schrijven.
Nadat u deze data in uw GfA- of C-programma heeft opgenomen, moet de sprite nog op het scherm gezet worden. Dit kan door middel van het kommando:

Sprite spritedata, x-positie,
y-positie (GfA)
Sprite-on(spritedata adres,
achtergronddata adres, x, y); (C)

Wissen van een sprite gaat via:
Sprite spritedata (GfA)
Sprite -off(achtergronddata adres); (C)

Wie een sprite over het beeldscherm laat bewegen, merkt dat deze fors knippert. Dit knipperen kan een beetje worden tegengegaan door voor elk sprite het kommando Vsync in te tikken. Maar de problemen kwamen pas goed op gang toen Peter Witvoet vier aaneengesloten sprites over het beeldscherm probeerde te jagen. Er bleven restjes achter en het knipperen van de sprites was niet meer om aan te zien. Het zag er naar uit dat de sprite voorgoed dood verklaard was. Maar...

## GEEN GEKNIPPER

...hier is een oplossing. Kijkt u nog eens in START 4 op pagina 40. Dat artikel gaat over het werken met meerdere schermen in C. Weliswaar is de listing in C, maar de techniek is ook heel goed in GfA-BASIC te gebruiken. De ST biedt standaard twee mogelijkheden om iets op een scherm te tekenen: direkt op het zichtbare scherm, of indirekt op een scherm wat u niet ziet. Dit biedt een aantal leuke mogelijkheden. In dit artikel gebruiken we deze techniek om een knippervrije sprite te krijgen. Allereerst reserveren we wat geheugen voor een tweede scherm. Dan installeren we dit scherm als nummer twee met behulp van de funktie Setscreen (= Xbios(5)). De eerste parameter van deze funktie is het adres van het tweede scherm, de tweede het adres van het eerste (zichtbare) scherm en de derde de resolutie. Daar de resolutie in dit geval niet mag veranderen, moet aan de derde parameter de waarde -1 worden meegegeven.
Als u nu iets gaat tekenen (bijvoorbeeld met behulp van een VDI-funktie) verschijnt dit op het tweede (onzichtbare) scherm. Nadat alles wat we wensen getekend is, maken we dit tweede scherm zichtbaar. Voor een mooie overgang is het nodig om eventjes te wachten totdat het scherm daadwerkelijk door de ST opgebouwd is. Gebruik hiervoor Vsync (xbios(37)).
Bij het omschakelen is het tweede scherm het zichtbare en het eerste scherm het onzichtbare geworden. Als u opnieuw iets gaat tekenen, verschijnt dit op het scherm wat u daarnet nog zag maar wat nu het tweede scherm is. Als u dit wilt laten zien, moet u weer omschakelen. Helaas staat alle informatie van de eerste keer tekenen nog steeds op het tweede beeldscherm. Voor het tekenen moeten we dus eerst wissen.
Door deze techniek toe te passen, verschijnt elke afbeelding zonder knipperen op het scherm. Misschien een aardig idee om 3D-Frames (START 5) zonder knipperen te programmeren?
Natuurlijk bestaan er ook andere wegen om een knippervrij beeld te krijgen. U kunt bijvoorbeeld met interrupts gaan werken. Zo'n interrupt zorgt ervoor dat de gebruiker niet kan tekenen terwijl de ST met iets anders bezig is.
Interrupts vragen echter om machinetaal programmering en zijn voor de meeste toepassingen niet zo geschikt. Ten tweede leent GfA-BASIC zich niet zo voor het programmeren van interrupts.

## GEEN RESTJES!

Een tweede probleem dat in het artikel van kollega Peter Witvoet werd aangesneden, was dat er restjes van sprites achterbleven. Gelukkig is dit vrij eenvoudig op te lossen, al staat het niet duidelijk in de GfA handleiding. Zorg ervoor dat de laatste sprite het eerste wordt weggehaald. Dit kan met erg veel sprites heel vervelend werken, maar voorlopig (zonder hardwarematige sprites of programmeertruuks) zit er niets anders op.
Listing 1 toont ons een GfA programma dat het bekende vliegtuigje uit 'Sprites en Spruiten' zonder knipperen en zonder restjes laat overvliegen. Eerst moet de gebruiker de snelheid van het vliegtuigje opgeven. Dit is de 'stapgrootte' waarmee het vliegtuigje vliegt.
Vervolgens wordt er ruimte voor een tweede scherm gemaakt. Omdat een beeldscherm altijd op een veelvoud van 256 moet staan, worden er 32255 bytes gereserveerd. Een Cls zorgt voor een schoon scherm. Dit kopieert het programma meteen naar scherm twee. Hierna maakt het twee series sprites aan (voor elk beeldscherm een serie). Een for/next-loop zorgt voor het bewegen van de sprite.

## C-BIBLIOTHEEK

Listing twee is nog interessanter. Het programma is gecompileerd met Megamax-C en werkt zowel in medium- als in highresolution. Als u het start, danst er een tweetal sprites over het scherm. Druk op de linker muisknop om het onvermijdelijke te laten gebeuren...
Het tweede voorbeeld is vrij lang, maar bevat voor elke C programmeur wel een aantal bruikbare routines. Als u een programma maakt dat zowel in de middelste als in hoge resolutie moet werken, doet u er verstandig aan om alle Y-koördinaten met twee te vermenigvuldigen. (Aangenomen dat de koördinaten voor medium zijn. Anders deelt u door twee.)
Het programma initialiseert eerst een aantal variabelen (o.a. voor GEM). SPRITE is een variabelentype waarvan elke variabele op een sprite struktuur wijst. De spritedata staat direkt onder de definitie van SPRITE.
BOINKJE is het geluid van een tegen een muur aanstuiterende bal.
Hierna volgen twee interessante assemblerroutines die het mogelijk maken om de sprites vanuit C te gebruiken. Direkt 'includen' van assembler kan alleen Megamax. Bij Lattice en DRI zullen deze routines een beetje aangepast en apart bij het gecompileerde programma gelinkt moeten worden.
Open- en Close-vwork zijn noodzakelijk voor elk C-programma dat gebruik maakt van de VDI tekenfunkties (gekenmerkt door de V voor elke funktienaam). Instal-s-scr(), dump-screen() en switch-screens() zijn bijna letterlijk overgenomen uit 'Snelheid is Schoonheid' in START 4. Instal-s-scr() maakt ruimte voor een tweede scherm, dump-screen() kopieert scherm één naar positie twee en switch-screens() verwisselt de twee beeldschermen.
Shift-c() is een routine die het programma in de VBL queue installeert. Deze routine wordt elke keer aangeroepen als de ST zijn beeldscherm opnieuw naar de monitor stuurt. Dit zorgt voor het grappige kleureneffekt in medium resolution. Officieel mag je niet zomaar je routine op adres 0x4D2 zetten, maar moet de programmeur vanaf adres 0x4CE zoeken tot hij een lege plaats aantreft. Het kan zijn dat een ander programma zijn routine toevallig ook al op adres 0x4D2 gezet had. Maar dit eventjes terzijde.
Startdemo() tekent het scherm, installeert shift-c() in de VBL queue en laat de sprite stuiteren. De afmetingen van de box waarin de sprite stuitert, kunnen naar eigen inzicht veranderd worden. Niets belet u om deze sprite bijvoorbeeld in een dialog-box te laten rondboinken.
Draw-st() tekent de sprites. Let op de parameterlijst: wat GfA automatisch doet (geheugen reserveren voor later achtergrondherstel) moet in C 'handmatig' gebeuren. Als tweede parameter geeft u sprite-on() het adres mee van een vrij stukje geheugen waar de ST zijn achtergrondinformatie kwijt kan. Dit adres

geeft u dan bij sprite-off() opnieuw door. De ST herstelt de achtergrond aan de hand van dit adres. Dit verklaart tegelijkertijd waarom de ST restjes achterlaat als de sprites in de verkeerde volgorde weggehaald worden. Een eerste sprite kan immers best tot de achtergrondinformatie van een tweede sprite behoren.
In monochroom moeten er minstens 74 (=37 integers) voor de achtergronden gereserveerd worden, in medium resolution minstens 138 en in low resolution minstens 266. Dit heeft te maken met het feit dat 1 kleur opslaan minder geheugen kost dan 4 of 16 kleuren. Houd er rekening mee, anders kan uw programma op schijnbaar onverklaarbare wijze crashen.
In de listing vindt u deze aanwijzing toegepast bij de funkties draw-st, del-st en do-sttje.
Calc-coor() en do-sttje() dragen zorg voor de stuiterende sprite.

## ● TIPS

- Maak een paar include files (bijvoorbeeld screens.c en sprite.c) waarin u respektievelijk de scherm- en de spriteroutines opneemt. U hoeft dan in het vervolg alleen nog maar include "sprite.c" in te tikken en de routines worden automatisch met het programma mee gecompileerd.
- Meerkleuren sprites zijn te suggereren door meerdere sprites over elkaar heen te laten vallen. Verander dan buiten de spritedata ook de voorgrond- en eventueel de achtergrondkleur.
- Zeer grote sprites zijn niet aan te bevelen. Gebruik in plaats hiervan rasterfunkties (Get en Put in GfA, vro-cpyfm() in C).
- De funktie Getrez() (Xbios(4)) geeft op elk gewenst moment de resolutie waarin gewerkt wordt:

0 = Low res
1 = Medium res
2 = High res

Aan de hand van de uitkomst kunt u het programma laten bepalen hoe de koördinaten omgerekend moeten worden om voor de gebruikte resolutie het gewenste resultaat op te leveren.
Een leuk geluidje en een aantal mooie sprites geven uw programma's iets extra. Experimenteer dus naar hartelust om de behandelde technieken in de vingers te krijgen. Probeer bijvoorbeeld eens een voorbij puffend treintje met een stoomfluit te programmmeren. Als dat u lukt, is het nog maar een kleine stap naar een nieuwe speelhaltopper.

*Egon Lubbers*

**LISTING 1**

```
' Sprites voorbeeld                         Egon Lubbers '87
'
Input "Snelheid van de sprites ";Speed
Dim Sp1$(4)
Dim Sp2$(4)
'
Scr2$=Space$(32256)                   ! tweede scherm
Scr2%=Varptr(Scr2$)                   ! op een veelvoud
Scr2%=Scr2%+256-(Scr2% Mod 256)       ! van 256.
'
Cls
Phys%=Xbios(2)                        ! Physbase
Bmove Phys%,Scr2%,32000               ! kopieer Phys naar Scr2
Gosub Lees.sprites
For X%=1 To 4
  Sp2$(X%)=Sp1$(X%)                   ! sprites kopieren
Next X%
'
Hidem
Ypos%=50
Void Xbios(5,L:Scr2%,L:Phys%,-1)
For Y%=546 To 1 Step -Speed*2         ! van rechts naar links
  Add Ypos%,1
  For X%=4 Downto 1
    Sprite Sp2$(X%)                   ! wis sprite 2 (in Scr2)
  Next X%
  For X%=1 To 4
    Sprite Sp2$(X%),Y%+X%*16,Ypos%! en teken sprite 2 op een
  Next X%                             ! nieuwe positie
  Void Xbios(5,L:Phys%,L:Scr2%,-1)! laat Scr2 maar zien
  Vsync                              ! wacht tot scherm geinstalleerd is
  For X%=4 Downto 1
    Sprite Sp1$(X%)                   ! wis sprite 1 (in Phys)
  Next X%
  For X%=1 To 4
    Sprite Sp1$(X%),Y%-Speed/2+X%*16,Ypos% ! teken sprite 1 (in Phys)
  Next X%
  Void Xbios(5,L:Scr2%,L:Phys%,-1)! toon Phys (en ga tekenen in Scr2)
  Vsync                               ! weer wachten op synchronisatie
Next Y%
Void Inp(2)                           ! wacht op een toets
Void Xbios(5,L:Phys%,L:Phys%,-1)      ! in Phys tekenen en Phys tonen
Showm
End
'
Procedure Lees.sprites
  For X%=1 To 4
    Sp1$(X%)=Mki$(8)+Mki$(8)+Mki$(0)+Mki$(0)+Mki$(1)
  Next X%
  For X%=1 To 16
    Read Sp1$
    For Y%=1 To 4
      X$="&X"+Mid$(Sp1$,16*Y%-15,16)
      Sp1$(Y%)=Sp1$(Y%)+Mki$(&HFFFF)+Mki$(Val(X$))
    Next Y%
  Next X%
Return
'
Data 0000000000000000000000000000000000000000000000000000000000000000
Data 0000000000000000000000000000000000000000000000000000000000111000
Data 0000000000000000000000000000000000000000000000000000000001111100
Data 0000000000000000000000000000000000000000000000000000000011111110
Data 0000000000000000000000000000000000000000000000000000000111111110
Data 0000000000000000000000000000000000000000000000000000001111111110
Data 0000000000001111111111111111111000000000000000000000000001111111110
Data 0000000000111000000000000111111000000000000000000000111111111110
Data 0000000001110000000000000011111100000000000000001111111111111110
Data 0000001111111111111111111111111111111111111111111111111111111110
Data 0001111111111111111111111111111111111111111111111111111111111110
Data 0111111111111111000000000000000000011111111111111000000000000000
Data 0111111111111111111111111111111111111111111111111111111111111110
Data 0000111111111111111111111111111111111111111111111111111111111000
Data 0000000111111111111111111111111111111111111111111111111111110000
Data 0000000000111111111111111111111111111111111111111111111111100000
```

**LISTING 2**

```
/* >>> SPRITE DEMO (MEDIUM en MONO) door Egon Lubbers (1987) <<< */

#include <osbind.h>
#define HIDEMOUSE  v_hide_c(handle)
#define SHOWMOUSE  v_show_c(handle, 1)
#define RED        0xFF8242

int handle, dummy, contrl[12], work_in[11], work_out[57],
   intin[128], intout[128], ptsin[128], ptsout[128], res = Getrez();
long *scr1, *scr2, *oldestscr;

typedef struct {
  int x_off,y_off,format,back_color,for_color,image[32];
          }SPRITE;

SPRITE SPST[2] = {0,0,0,0,1,
      {0,0,8191,0,8198,8185,16390,16377,32780,32755,36855,28672,
      36832,28672,32784,32736,16392,16368,8192,8184,8132,56,
      16324,56,16388,16376,32776,32752,32784,32736,65504,0},
            0,0,0,0,1,
      {0,0,65520,0,16,65504,16,65504,32,65472,58304,7168,8704,
      7168,8704,7168,8704,7168,8704,7168,8704,7168,8704,7168,
      8704,7168,8704,7168,8704,7168,15872,0}};  /* sprite data */

char BOINKJE[] = {6,21, 7,247, 8,16, 11,128, 12,25, 13,3, 255,0};

sprite_on(sp_def,sp_save,x_sp,y_sp)
long sp_def,sp_save;
int x_sp,y_sp;
{
  asm{
      MOVEM.L   A2/A6,-(A7)
      MOVE.W    x_sp(A6),D0
      MOVE.W    y_sp(A6),D1
      MOVE.L    sp_def(A6),A0
      MOVE.L    sp_save(A6),A2
      DC.W      0xA00D                        ; A00D : draw sprite
      MOVEM.L   (A7)+,A2/A6
    }
}

sprite_off(sp_save)
long sp_save;
{
  asm{
      MOVE.L    A2,-(A7)
      MOVE.L    sp_save(A6),A2
      DC.W      0xA00C                 ;    A00C : undraw sprite
      MOVE.L    (A7)+,A2
    }
}

open_vwork()
{
  int i;

  appl_init();
  handle = graf_handle(&dummy, &dummy, &dummy, &dummy);
  for (i = 0; i < 10; work_in[i++] = 1);
  work_in[10] = 2;
  v_opnvwk(work_in, &handle, work_out);
}

close_vwork()
{
  v_clsvwk(handle);
  appl_exit();
}

instal_s_scr()     /* geheugen reserveren voor 2de scherm */
{
  long mm;
```

```
  if( ! (mm = Malloc(32255L))) {
    form_alert(1,"[3][Run out of memory !][Cancel]");
    Pterm0(); /* terug naar desktop... */
  }
  oldestscr = scr1 = (long *) Physbase();
  scr2 = (long *) ((mm+0xFF) & 0xFFFF00L);
}

dump_screen()
{
  register int i;
  register long *dump_scr = scr2, *old_scr = scr1;

  for (i = 0; i < 8000; ++i)
    *dump_scr++ = *old_scr++;
}

switch_screens()
{
  long *sw_it = scr1;

  scr1 = scr2; scr2 = sw_it;
  Setscreen(scr2,scr1,-1); /* verwissel schermen */
  xbios(37);
}

shift_c()
{
  ++ *(int *) RED;      /* rood laten verlopen ... */
}

main()
{
  open_vwork();
  HIDEMOUSE;
  instal_s_scr();
  spritedemo();
  SHOWMOUSE;
  Setscreen(oldestscr, oldestscr, -1);
  close_vwork();
}

draw_rect(interior, style, pxy)
int style, interior, pxy[4];
{
  vsf_style(handle,style);
  vsf_interior(handle, interior);
  v_bar(handle, pxy);
}

spritedemo()
{
  int pxy[4], old_color;
  long user;

  if (res == 1){                   /* medium resolution ?          */
    user = Super(0L);              /* systeemvariabelen : Super ! */
    *(long *) 0x4D2 = (long) shift_c;/* over 0x4D2 (normaal vrij)   */
    old_color = * (int *) RED;     /* shift_c in VBLqueue en red rood*/
    Super(user);                   /* een weer terug in user mode */
  }
  pxy[0] = 120; pxy[1] = res * 25; pxy[2] = 520; pxy[3] = res * 175;
  graf_growbox(3,3,15,15,pxy[0],pxy[1],pxy[2]-pxy[0],pxy[3]-pxy[1]);
  draw_rect(0, 0, pxy);
  pxy[0]++; pxy[1] ++; pxy[2] --; pxy[3] --;
  draw_rect(0, 0, pxy);
  pxy[0] += 4; pxy[1] += 4; pxy[2] -= 4; pxy[3] -= 4;
  draw_rect(2, 9, pxy); /* baksteentjes */
  pxy[0] += 35; pxy[1] += res * 17; pxy[2] -= 35; pxy[3] -= res * 17;
  draw_rect(0, 0, pxy);
  vst_height(handle, 6, &dummy, &dummy, &dummy, &dummy);
  vst_color(handle, 2);        /* rood               */
  v_gtext(handle, 210, res * 140, "SPRITE DEMO door Egon Lubbers");
  vst_color(handle, 1);
  vst_height(handle, 13, &dummy, &dummy, &dummy, &dummy);

  do_sttje(120, res * 25, 400, res * 150);

  if (res == 1){
    user = Super(0L);
    *(long *) 0x4D2 = 0L;
    *(int *) 0xFF8242 = old_color;       /* en er weer uit ! */
    Super(user);
  }
  evnt_timer(3000,0);                  /* wacht even */
  graf_shrinkbox(3, 3, 15, 15, pxy[0], pxy[1], pxy[2]-pxy[0], pxy[3]-pxy[1]);
}

draw_st(stx,sty,aback)
int stx,sty,aback[2][69];
{
  sprite_on(&SPST[0],&aback[0][0],stx,sty);
  sprite_on(&SPST[1],&aback[1][0],stx+16,sty);
}

del_st(aback)
int aback[2][69];
{
  sprite_off(&aback[1][0]);    /* laatste als eerste */
  sprite_off(&aback[0][0]);
}

cal_coor(xb, yb, wb, hb, stx, sty)
int xb, yb, wb, hb, *stx, *sty;
{
  static int xdir = 2, ydir = 1;
  int ymve;

  *stx += xdir;
  ymve = (*sty-yb)/((yb+hb)/16);
  *sty += (ydir * ymve);

  if (ymve == 0)          {*sty += 1; ydir = 1;}  /* boven */
  if (*stx >= xb+wb-32) {xbios(32,&BOINKJE); xdir =- 2;} /* rechts */
  if (*stx <= xb)        {xbios(32,&BOINKJE); xdir = 2;}  /* links  */
  if (*sty >= yb+hb-28) {xbios(32,&BOINKJE); ydir =- 1;} /* onder  */
}

do_sttje(xb, yb, wb, hb)
int xb, yb, wb, hb;
{
  int stx, sty, mbutton, dummy, xart = 320, yart = res * 75;
  int back1[2][69],back2[2][69]; /* 2 * achtergrond voor 2 sprites */
                                 /* indien alleen mono ipv 69  37  */
  stx = xb + (wb/2)-16;          /* netjes in 't midden */
  sty = yb + (hb/2)-16;
  vst_effects(handle, 4);        /* cursief                 */
  v_gtext(handle, xart, yart, "ART");
  dump_screen();
  Setscreen(scr2, scr1, -1);
  draw_st(stx, sty, back1);
  switch_screens();

  do{
    cal_coor(xb, yb, wb, hb, &stx, &sty);
    draw_st(stx, sty, back2);
    switch_scr();
    cal_coor(xb, yb, wb, hb, &stx, &sty);
    del_st(back1);
    draw_st(stx, sty, back1);
    switch_scr();
    vq_mouse(handle, &mbutton, &dummy, &dummy);
    del_st(back2);
  } while( ! mbutton);
  xart -= 23;
  yart -= 14;
  while(stx != xart || sty != yart){           /* ST naar ART */
    stx += stx < xart ? 1 : (stx == xart ? 0 : -1);
    sty += sty < yart ? 1 : (sty == yart ? 0 : -1);
    draw_st(stx, sty, back2);
    switch_scr();
    stx += stx < xart ? 1 : (stx == xart ? 0 : -1);
    sty += sty < yart ? 1 : (sty == yart ? 0 : -1);
    del_st(back1);
    draw_st(stx, sty, back1);
    switch_scr();
    del_st(back2);
  };
  dump_screen(); /* voor de afwerking */
  Setscreen(scr1, scr1, -1);
}
```

# Maak 'n MAKE

**Zoals u al in START 4 heeft kunnen lezen, is een make-utility voor iedere serieuze programmeur een onmisbaar bezit. Mark Boon heeft enige tijd gewerkt met de make die in de Megamax-shell is opgenomen, maar dat beviel hem niet zo. Hij zette zijn ergernis om in daden en is graag bereid het resultaat met u te delen.**

## TWEE EXTRA'S

De make die bij Megamax wordt geleverd, is helaas slechts een aftreksel van de standaard: het Unix make. Bovendien is de Megamax make lang niet zo handig. Er is echter iets aan de Megamax make dat bij mijn weten niet in andere make-utilities zit: de shell start, na een fout bij het compileren, automatisch de editor op met de file waar de fout in zit. Bovendien deponeert hij de file met foutmeldingen in buffer twee. Dit werkt natuurlijk alleen met een editor die meerdere files tegelijk kan verwerken.
Daar ik zowel deze laatste mogelijkheid als een aantal van de faciliteiten van Unix make begeerde, zat er dus niets anders op dan zelf een make te maken die volledig aan mijn wensen voldeed. Behalve het gebruikelijke principe moest deze make twee extra's bevatten. Ten eerste de al genoemde faciliteit om de editor aan te roepen met de file waar de fout in zit, zoals de Megamax-shell dat doet. Het moest dus mogelijk zijn om in de makefile aan te geven wat er moet gebeuren als een programma een foutmelding geeft.

Ten tweede wilde ik graag de mogelijkheid hebben om, tussen de compile- en linkopdrachten door, programma's van schijf te verwijderen om diskruimte vrij te maken, of om na het compileren de 'codeimprover' die bij Megamax zit aan te roepen. Hiervoor moet make dus meerdere kommando's achter elkaar uit kunnen voeren.
Het programma MAKE.TTP, voldoet aan de bovenstaande eisen. Het kent geen makro's en is daardoor niet zo omvangrijk en veelzijdig als de Unix make. Dat is ook niet nodig: het ging mij om een kompakt programma dat zijn werk beter zou doen dan de Megamax make.

## INSTALLATIE

Het hier afgedrukte bestand PROGRAMS.MAK is een voorbeeld van een makefile. De naam van de dubbele punt stelt de 'target' voor. De namen er achter bepalen de files waar de target van afhankelijk is. Dit is konform het gebruik van Unix make. Vervolgens staan er één of meer regels die met een '@' beginnen. Deze regels worden één voor één uitgevoerd als dat nodig blijkt te zijn. Als er tijdens deze kommando's een fout optreedt, wordt de eerste regel die met een '#' begint opgezocht en het kommando wat er achter staat uitgevoerd. Staat er een

**MAKE.TTP**

```
#include <stdio.h>
#include <osbind.h>

#define getchar() ((char) bios(2,2))
#define puts(strptr) gemdos(0x9,strptr)

extern long gemdos();

char dtabuff[44];
char *filebuf;
long filesize;

main(argc,argv)
int  argc;
char     **argv;
{
     char program[20];
     register int fp1,i=0;
     int fc=0;

     if (argc != 2) error("Usage: make.ttp 'makefile'");

     puts(" MAKE... by Mark Boon   Amsterdam, 1987\n\n\r");

     /* bepaal de grootte van de makefile */
     /* alloceer vervolgens genoeg geheugen en lees de file in */
     Fsetdta(dtabuff);
     if (Fsfirst(argv[1],0x7) == -33)
     {
          puts(argv[1]);
          error(" not found\n\r");
     }
     filesize=((unsigned int) dtabuff[26])*256*256*256+
                                   ((unsigned int) dtabuff[27])*256*256+
                                   ((unsigned int) dtabuff[28])*256+
                                          (unsigned int) dtabuff[29];

     filebuf=(char *)Malloc(filesize);
     fp1=Fopen(argv[1],0);
     Fread(fp1,filesize,filebuf);
     Fclose(fp1);

     /* Zoek de programmanaam en ga dit programma maken */
     skip(&fc);
     while (filebuf[fc] != ':') program[i++]=filebuf[fc++];
     program[i]='\0';
     make(program,0);
     exxit();
}

make(file,fc)
char *file;
int fc;
{
 register int i,time,date,found=0,uptodate=1;
 char nextfile[127],commando[127],tail[127];

 do /* zoek de target */
 {
  skip(&fc);
  i=0;
  while (file[i] != '\0')
    if (file[i++] != filebuf[fc++]) break;
  if (file[i] == '\0' && filebuf[fc] == ':')
  {
   found=1;
   fc++;
  }
 }
 while (!found && fc <= filesize);

 if (!found) /* Deze file is nergens van afhankelijk */
  return;

 do /* maak eerst recursief de files waarvan 'file' afhankelijk is */
 {
  skip(&fc);
  if (fc >= filesize || filebuf[fc] == '@' || filebuf[fc] == '#') break;
  i=0;
  while (filebuf[fc] := ' ' && filebuf[fc] != '\n' && filebuf[fc] != '\r')
     nextfile[i++]=filebuf[fc++];
  if (i == 0) break;
  nextfile[i]='\0';
  make(nextfile,fc);

  if (Fsfirst(file,0x7) == -33)
  /* als 'file' niet op schijf staat dan is deze zowiezo niet up to date */
     uptodate=0;
  else
  /* als 'file' wel op schijf staat dan wordt gekeken of deze afhankelijk is
  van een file met een latere tijd of datum */
  {
   time=(unsigned int) dtabuff[22]*256+(unsigned int) dtabuff[23];
   date=(unsigned int) dtabuff[24]*256+(unsigned int) dtabuff[25];
```

```
    if (Fsfirst(nextfile,0x7) != -33)
    {
     if (date < (unsigned int) dtabuff[24]*256+(unsigned int) dtabuff[25]
      || (date == (unsigned int) dtabuff[24]*256+(unsigned int) dtabuff[25]
      && time < (unsigned int) dtabuff[22]*256+(unsigned int) dtabuff[23]))
        uptodate=0;
    }

    /* Bij grote programmas kan dit beter weggelaten worden omdat je dan
    C-files die toch nooit veranderen op een andere schijf kunt zetten. */
    else
    {
     puts(nextfile);
     error(" not found\n\r");
    }
   }
  }
  while (1);

  if (!uptodate)
  {
   while (filebuf[fc++] == '@')
   {
    skip(&fc);
    i=0;
    while (filebuf[fc] != '\n' && filebuf[fc] != '\r'
    && filebuf[fc] != ' ' && fc <= filesize)
       commando[i++]=filebuf[fc++];
    commando[i]='\0';
    skip(&fc);
    i=1;
    while (fc <= filesize && filebuf[fc] != '@' && filebuf[fc] != '#')
    {
     if (filebuf[fc] != '\n' && filebuf[fc] != '\r')
        tail[i++]=filebuf[fc];
     fc++;
    }
    tail[i]='\0';
    tail[0]=i;

    if (Pexec(0,commando,tail,0L))
    {
     while (filebuf[fc] != '#' && fc < filesize) fc++;
     if (filebuf[fc++] == '#')
     {
      i=0;
      while (filebuf[fc] == ' ') fc++;
      while (filebuf[fc] != '\n' && filebuf[fc] != '\r'
      && filebuf[fc] != ' ' && fc <= filesize)
         commando[i++]=filebuf[fc++];
      commando[i]='\0';
      skip(&fc);
      i=1;
      while (filebuf[fc] != '\n' && filebuf[fc] != '\r' && fc <= filesize)
         tail[i++]=filebuf[fc++];
      tail[i]='\0';
      tail[0]=i;
      if (commando[0] != '\0')
      {
       if (Pexec(0,commando,tail,0L))
       {
        puts(" error-command failed\n\r");
        exxit();
       }
      }
      fc=filesize;
     }
     exxit();
    }
   }
  }
 }

 skip(i)       /* sla spaties, newlines etc. over */
 int *i;
 {
                 while (filebuf[*i] <= ' ' && *i < filesize) ++(*i);
 }

 /* print een foutmelding en verlaat het programma na een toetsaanslag */
 error(strpoi)
 char *strpoi;
 {
     puts(strpoi);
     getchar();
     exit(1);
 }

 exxit()
 {
     puts("\n Press a key to leave MAKE...\n\r");
     getchar();
     exit(0);
 }
```

lege regel achter het hekje, dan gebeurt er niets.
Make is vanuit de desktop het beste te gebruiken als u al uw makefiles laat eindigen op de extensie MAK. Vervolgens klikt u een keer op MAKE.TTP en kiest 'Install Application'. Daar vult u ook 'MAK' in, zodat u vervolgens alleen nog maar met de muis op uw makefile hoeft te klikken om hem naar MAKE.TTP te sturen.
Het is ook mogelijk om make voor andere talen dan C te gebruiken. Een andere optie is een makefile te maken die geen programma, maar een funktiebibliotheek up-to-date maakt.
Aangezien make van de datum en de tijd afhankelijk is, moeten deze steeds juist ingesteld staan. Ik vind het controlpanel daar niet zo handig voor, dus heb ik ook hier een programma voor gemaakt. TIME.TOS is niet mooi, maar wel heel praktisch.
Tot slot vindt u een listing van RM.TTP. Met dit utility kunt u een bestand van schijf verwijderen. Een voorbeeld van een toepassing ziet u in PROGRAMS.MAK.
Ik hoop dat make voor u net zo'n plezierig hulpmiddel zal blijken te zijn als het voor mij is.

*Mark Boon*

PROGRAMS.MAK

```
programs.prg: prog1.o prog2.o
@ rm.ttp programs.prg
@ \megamax\kernel\mmlink.ttp prog1.o prog2.o -o programs.prg
#

prog1.o: prog1.c header.h
@ \megamax\kernel\mmcc.ttp prog1.c
# editor.prg prog1.c errors.out

prog2.o: prog2.c header.h
@ \megamax\kernel\mmcc.ttp prog2.c
# editor.prg prog2.c errors.out
```

TIME.TOS

```
#include <stdio.h>
#include <osbind.h>

#define puts(a) gemdos(0x9,a)
#define gets(a) gemdos(0xa,a)
/* Dit programma vraagt de datum en de tijd van het toetsenbord. */
/* 15-2-1987  Mark Boon */

main()
{
        int day,month,year;
        int minutes,hours;
  char input[10];

  input[0]=7;
      do
      {
        puts("\n\rEnter date:\n\r");
        puts("DDMMYY\r");
        gets(input);
        day=10*(input[2]-'0')+input[3]-'0';
        month=10*(input[4]-'0')+input[5]-'0';
        year=10*(input[6]-'0')+input[7]-'0';
      }
      while(year<80 || month<1 ||     month>12 || day<1 || day>31);
      Tsetdate(((year-80)<<9)|(month<<5)|day);

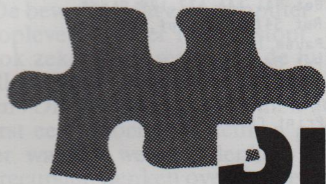
  input[0]=5;
      do
      {
        puts("\n\rEnter time:\n\r");
        puts("HHMM\r");
        gets(input);
        hours=10*(input[2]-'0')+input[3]-'0';
        minutes=10*(input[4]-'0')+input[5]-'0';
      }
      while(hours<0 || hours>23 || minutes<0 ||        minutes>59);
      Tsettime((hours<<11)|(minutes<<5));
}
```

RM.TTP

```
#include <osbind.h>

main(argc,argv)
int argc;
char **argv;
{
      while (argc!=0)
            Fdelete(argv[argc--]);
}
```

***Het eerste en lange tijd zelfs het enige spelletje dat voor een 68000-processor bestond, was een puzzelspelletje dat als "accessory"-programma in het Macintosh besturingssysteem was opgenomen.***
***Sedertdien is een lange weg afgelegd, en met enige fierheid mag worden gezegd dat de komst van de ATARI-ST in niet geringe mate heeft bijgedragen tot de grote vlucht die spelletjes-software voor de 68000-systemen heeft genomen.***
***Met het Puzzler-programma voert Karl Moens ons "back to the roots".***



# PUZZLER

## RESOLUTIE

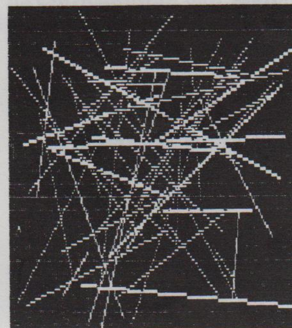
Puzzler is geschreven in GfA-Basic versie 2.0 en kan eventueel worden gecompileerd met behulp van de GfA-Basic Compiler. Het programma loopt alleen op een kleurenmonitor in medium-resolutie.
Na het starten van het programma wordt links bovenaan een voorbeeld van de voltooide puzzel getoond. De puzzelstukken bevinden zich onderaan het scherm. De speler moet de muiscursor op het gewenste stuk plaatsen en, terwijl hij de linker muisknop ingedrukt blijft houden, het stuk naar zijn juiste plaats in het rechter kader trekken. Hierna kan de muisknop losgelaten worden.
Tijdens het spelen is er een klok in beeld, zodat de speler zijn puzzelvaardigheid kan toetsen.
Om het spel te beëindigen of een nieuwe puzzel te beginnen, volstaat het op de STOP-knop te drukken.

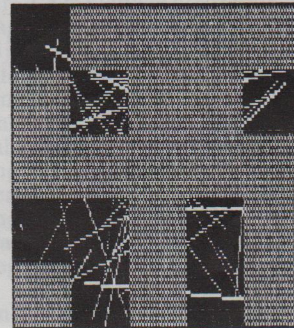
## RANDOM PUZZELS

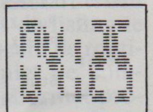
Puzzler bestaat uit drie grote delen: een initialisatieprocedure, de uitvoering van het programma zelf in de vorm van een oneindige 'Do ... Loop' en de reeks van procedures die wordt aangeroepen tijdens de initialisatie en gedurende de loop van het programma.
In het initialisatie-deel (Procedure Initia) wordt de nodige geheugenruimte gereserveerd, het scherm opgebouwd, de klok op nul gesteld en de puzzel getekend en in stukken 'geknipt'.
Er zijn twee verschillende puzzels mogelijk: een puzzel die opgebouwd is uit 'random' lijnen of een puzzel die bestaat uit random rechthoeken.
Wie na een tijdje op deze puzzels is uitgekeken, kan op zeer eenvoudige wijze puzzels van eigen ontwerp toevoegen. Het volstaat in de procedure Initia de regel
On Random(2)+1 Gosub Rand-lines, Rand-boxes aan te vullen met de eigen procedure en natuurlijk het cijfer twee te vervangen door het aantal procedures waaruit thans kan worden gekozen. Zolang de eigen tekening binnen het vierkant dat gevormd wordt door de overstaande hoekpunten (9,4) en (210,105) blijft, kan er niets mis gaan.
Wie schrijft er trouwens een procedure die toelaat een (stuk van) een DEGAS-tekening in te lezen als puzzel?
Het in stukken knippen van de puzzel gebeurt in de procedure Maak-stukken. De Get-funktie verdeelt de puzzel in vijfentwintig stukken en bewaart hem in de Stuk$-array. De Stuk-lijst%-array houdt bij waar elk stuk zich bevindt.

## PSEUDO SPRITES

Na eventueel de klok in de rechter bovenhoek van het scherm aangepast te hebben (via de procedure Klok), wordt getest of de linker muisknop werd ingedrukt. Als dit het geval is, kijkt het programma of de muiscursor op een puzzelstuk staat, dan wel de STOP-knop werd ingedrukt (procedure Check-buttons).
Als er een geldig puzzelstuk werd aangewezen, verandert de muiscursor van een pijl in een hand (Defmouse 4) en wordt de procedure Move-stuk aangeroepen. Deze blijft aktief zolang de linker muisknop ingedrukt blijft.
De procedure Move-stuk is een 'pseudo sprite' procedure die enige extra aandacht verdient. Deze techniek kan ook in andere programma's nuttig zijn. Daar de puzzelstukken 40 x 20 pixels groot zijn, was het zo goed als onmogelijk de sprite-funktie te gebruiken. De struktuur van een sprite is trouwens niet compatibel met het formaat van een met Get opgehaalde tekening, zodat het gebruik van sprites zeer omslachtig zou zijn.



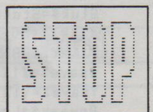













Het principe van de pseudo-sprite is zeer eenvoudig. Eerst wordt het scherm hersteld door de in Bewaar$ opgeslagen tekening terug te plaatsen. Vervolgens slaat het programma het gedeelte van het scherm dat zo meteen door de 'sprite' zal worden overschreven in deze variabele op. Tenslotte wordt de sprite op de nieuwe plaats ge'Put'.
Om het hinderlijke flikkeren van het scherm te voorkomen, dient een grafische bewerking vooraf te gaan door een Vsync-instruktie.
Als de sprite niet moet worden bewogen, is het nutteloos het scherm te bewaren en te herstellen. De test
If Abs(X%-X1%) >1
Or Abs(Y%-Y1%) >1
heeft tot gevolg dat de sprite slechts wordt hertekend als een beweging van meer dan 1 pixel in horizontale of vertikale richting werd waargenomen. (X% en Y% zijn de huidige coördinaten van de sprite, X1% en Y1% zijn de vorige coördinaten). De tekeningen ogen hierdoor stabieler en het programma loopt sneller.
Zodra de linker muisknop wordt losgelaten, zal de procedure Check-place nagaan of het puzzelstuk nu op de juiste plaats staat. Hierbij zal een stuk dat de halve breedte en/of hoogte naast de exakte plaats is losgelaten toch nog goed terecht komen. Als het stuk fout werd geplaatst, stuurt puzzler het terug naar zijn beginpositie.

Telkens als een stuk juist wordt geplaatst, verhoogt het programma de variabele Aantal% met 1 (Inc-instruktie). In een vorige versie van PUZZLER werd deze variabele getest om vast te stellen of de puzzel voltooid was. Op dat ogenblik verscheen dan een Alertbox met een mededeling dat het spel was afgelopen. Het bleek echter dat sommige gebruikers dit storend vonden (de Alertbox bedekt een deel van de tekening) en zodoende werd deze test verwijderd. De verwijzingen naar Aantal% bleven echter behouden om eventuele latere uitbreidingen die hiervan gebruik zouden maken niet in de weg te staan (bijvoorbeeld het bijhouden van de tijd die nodig was om de puzzel samen te stellen voor een lijst met 'high scores').
Het programma kan nu alleen worden verlaten door op de STOP-knop te klikken, waarna de mogelijkheid wordt geboden om hetzij een nieuwe puzzel te beginnen, hetzij het programma definitief te verlaten.

Dit programma kan zonder meer gecompileerd worden met de GfA-Basic compiler. Alleen de 'Setcolor i,r,g,b'-funktie gaf moeilijkheden, zodat ik ze heb vervangen door 'Setcolor i,n'. Dit komt op hetzelfde neer, maar laat helaas niet zo duidelijk zien hoe de kleur is samengesteld.

```
 If Xbios(4)<>1 Then
   Alert 3,"This program needs|          M =>
 EDIUM|    resolution.",1,"Sorry!",Co%
   End
 Endif
 Gosub Donker
 Opnieuw:
 Gosub Initia
 Do
   Mouse Muis_x,Muis_y,Muis_klik
   Gosub Klok
   If Muis_klik=1
     Volgnummer%=0
     Gosub Get_volgnummer(Muis_x,Muis_y)
     If Volgnummer%>0
       Gosub Get_stuk(Volgnummer%)
       Defmouse 4
       Repeat
         Gosub Klok
         Gosub Move_stuk
       Until Mousek=0
       Defmouse 0
       Gosub Check_place
     Else
       Gosub Check_buttons
     Endif
   Endif
   Exit If Opnieuw%=1
 Loop
 Clear
 Goto Opnieuw

 Procedure Klok
   Tijd%=(Timer-Start%) Div 200
   Minuten%=Tijd% Div 60
   Seconden%=Tijd% Mod 60
   If Minu%<>Minuten% Or Seco%<>Seconden%
     Minu1$=Chr$(Asc(Left$(Right$("0"+ =>
 Str$(Minuten%),2),1))-32)
     Minu2$=Chr$(Asc(Right$(Str$(Minut =>
 en%),1))-32)
     Seco1$=Chr$(Asc(Left$(Right$("0"+ =>
 Str$(Seconden%),2),1))-32)
     Seco2$=Chr$(Asc(Right$(Str$(Secon =>
 den%),1))-32)
     Tijd$=Minu1$+Minu2$+":"+Seco1$+Se =>
 co2$
     Deftext 3,0,0,32
     Vsync
     Text 550,50,Tijd$
     Minu%=Minuten%
     Seco%=Seconden%
   Endif
 Return

 Procedure Check_place
   If X1%>307 And X1%<480 And Y1%>2 An =>
 d Y1%<95
     X_puz%=(X1%-290) Div 40
     Y_puz%=(Y1%+5) Div 20
     ' Print At(1,1);X_puz%;Y_puz%;
     Nummer%=X_puz%+1+Y_puz%*5
     If Nummer%=Stuk_lijst%(Volgnummer%)
       Put X1%,Y1%,Bewaar$,3
       Put X_puz%*40+310,Y_puz%*20+5,S =>
 tuk$
       Stuk_lijst%(Volgnummer%)=0
       Inc Aantal%
     Else
       Put X1%,Y1%,Bewaar$,3
       Put Xsave%,Ysave%,Stuk$
     Endif
   Else
     Put X1%,Y1%,Bewaar$,3
     Put Xsave%,Ysave%,Stuk$
   Endif
 Return
 '
 Procedure Get_stuk(Volgnummer%)
   X_stuk%=((Stuk_lijst%(Volgnummer%)- =>
 1) Mod 5)+1
   Y_stuk%=((Stuk_lijst%(Volgnummer%)- =>
 1) Div 5)+1
   Stuk$=Stuk$(X_stuk%,Y_stuk%)
   X1%=X%*50-41
   Y1%=Y%*28+90
   Xsave%=X1%
   Ysave%=Y1%
   Bewaar$=Leeg$
 Return

 Procedure Move_stuk
   X%=Mousex-20
   Y%=Mousey-10
   If X%>0 And X%<600 And Y%>0 And Y%<180
     If Abs(X%-X1%)>1 Or Abs(Y%-Y1%)>1
       Vsync
       Put X1%,Y1%,Bewaar$,3
       Get X%,Y%,X%+40,Y%+20,Bewaar$
       Vsync
       Put X%,Y%,Stuk$,3
       X1%=X%
       Y1%=Y%
     Endif
   Endif
```

```
 Return
 '
 Procedure Donker
   Setcolor 0,0
   Setcolor 3,1365
   Setcolor 2,1792
   Setcolor 1,112
 Return
 '
 Procedure Plaats_stukken
   For Y%=1 To 3
     For X%=1 To 10
       Volgnummer%=X%+(Y%-1)*10
       X_stuk%=((Stuk_lijst%(Volgnumme =>
 r%)-1) Mod 5)+1
       Y_stuk%=((Stuk_lijst%(Volgnumme =>
 r%)-1) Div 5)+1
       Color 1
       If Stuk_lijst%(Volgnummer%)<>0
         Put X%*50-40,90+Y%*28,Stuk$(X =>
 _stuk%,Y_stuk%)
       Endif
     Next X%
   Next Y%
 Return
 '
   ' Mousepositie => Volgnummer stuk_lij =>
 st ; 0 = ERROR
 Procedure Get_volgnummer(Muis_x,Muis_y)
   If 9<=Muis_x And Muis_x<=501 And 11 =>
 7<=Muis_y And Muis_y<=195
     X%=(Muis_x+41) Div 50
     Y%=(Muis_y-89) Div 28
     Volgnummer%=X%+(Y%-1)*10
     If Stuk_lijst%(Volgnummer%)=0
       Volgnummer%=0
     Endif
   Else
     Volgnummer%=0
   Endif
 Return
 '
 Procedure Maak_stukken
   For X%=1 To 5
     For Y%=1 To 5
       Get X%*40-30,Y%*20-15,X%*40+9,Y =>
 %*20+4,Stuk$(X%,Y%)
     Next Y%
   Next X%
 Return
 '
 Procedure Door_elkaar
   Plaats%=Random(30)
   For T%=1 To 25
     Stuk_lijst%(Plaats%+1)=T%
     Plaats%=(Plaats%+7) Mod 30
   Next T%
 Return
 '
 Procedure Rand_lines
   For I%=1 To 50
     Colr=Random(3)+1
     If Colr=1
       Colr=0
     Endif
     Color Colr
     Line 10+Random(200),5+Random(100) =>
 ,10+Random(200),5+Random(100)
   Next I%
 Return
 '
 Procedure Rand_boxes
   For I%=1 To 7
     Colr=Random(3)+1
     If Colr=1
       Colr=0
     Endif
```

```
     Color Colr
     Xlh%=Random(100)
     Xrh%=99+Random(100)
     Ylh%=Random(50)
     Yrh%=49+Random(50)
     For T%=1 To 5+Random(5)
       Box 10+Xlh%,5+Ylh%,10+Xrh%,5+Yrh%
       Add Xlh%,Random(15)-7
       Xlh%=Max(Min(Xlh%,199),0)
       Add Xrh%,Random(15)-7
       Xrh%=Max(Min(Xrh%,199),0)
       Add Ylh%,Random(15)-7
       Ylh%=Max(Min(Ylh%,99),0)
       Add Yrh%,Random(15)-7
       Yrh%=Max(Min(Yrh%,99),0)
     Next T%
   Next I%
 Return
 '
 Procedure Check_buttons
   If Muis_x<240 And Muis_y<120
     Alert 0,"      P U Z Z L E |        =>
 v1.0|c 1987 By Karl MOENS",1,"Continu =>
 e",Co%
   Endif
   If 540<Muis_x And Muis_x<638 And 14 =>
 5<Muis_y And Muis_y<180
     Get 541,146,637,179,Stop$
     Put 541,146,Stop$,12
     Pause 15
     Put 541,146,Stop$
     Gosub Einde
   Else
     Print Chr$(7);
   Endif
 Return
 '
 Procedure Einde
   Alert 2,"Another puzzle?",1,"Yes|No =>
 ",Co%
   If Co%=2
     Setcolor 0,1911
     Setcolor 3,0
     Setcolor 1,1792
     Setcolor 2,112
     Cls
     End
   Else
     Opnieuw%=1
   Endif
 Return
 '
 Procedure Initia
   Dim Stuk$(5,5),Stuk_lijst%(30)
   Cls
   Get 1,1,43,23,Leeg$
   Gosub Door_elkaar
   Color 1
   Deffill 1,2,8
   Pbox 9,4,210,105
   Deffill 1,2,4
   Pbox 309,4,510,105
   On Random(2)+1 Gosub Rand_lines,Ran =>
 d_boxes
   Gosub Maak_stukken
   Gosub Plaats_stukken
   Color 3
   Box 540,20,638,55
   Box 540,145,638,180
   Deftext 3,17,0,32
   Text 550,174,80,"STOP"
   Aantal%=0
   Start%=Timer
   Minu%=-1
   Gosub Klok
   Gosub Donker
 Return
```

**Met Logo kunt u uw ST laten denken dat hij denkt. De Logo listing van deze keer is er een die u laat zien dat Logo nauwe relaties heeft met het onderzoek naar artificiële, ofwel kunstmatige intelligentie. Met deze listing kunt u uw ST een probleem laten oplossen, zodat u zich af zou kunnen vragen...**

# KUNNEN COMPUTERS DENKEN?

```
TO START
HT ERNS
BEGIN
MAKE "LENGTE INT (100 / :AANTAL)
MAKE "HOOGTE INT (150 / :AANTAL)
FULLSCREEN CLEARSCREEN
MAKE "BEURT 0
MAKE "ATOP -100 MAKE "BTOP -100 MAKE  =>
"CTOP -100
MAKE "GFILL "FALSE BOX [-150 140 300 20]
SETTEXT 17
PU SETPOS [-95 144] PD TT [De Torens  =>
van Hanoi]
SETTEXT 0
(LOCAL "XPOS "YPOS "NUM "R "L "MAAT)
MAKE "R -110 MAKE "L 65
BOX [-150 -150 300 50]
MAKE "XPOS -150 MAKE "YPOS -100 MAKE  =>
"NUM :AANTAL
LABEL "PLAATS
IF EQUALP :NUM 0 [GO "EIND]
MAKE "MAAT LIST (:LENGTE * :NUM) :HOOGTE
STEEN :NUM :XPOS :YPOS 1
MAKE "NUM :NUM - 1
MAKE "XPOS :XPOS + (:LENGTE / 2)
MAKE "YPOS :YPOS + :HOOGTE
MAKE "ATOP :YPOS
GO "PLAATS
LABEL "EIND
MAKE "GFILL "FALSE
IF :R > 90 [GO "HANOI]
BOX SE :R [-135 20 20]
PU SETPOS LIST :R + 6 -130 PD
TT CHAR :L
MAKE "R :R + 100
MAKE "L :L + 1
GO "EIND
LABEL "HANOI
MAKE "NUM :AANTAL
ZET :NUM "A "C "B
EINDE
END

TO VERPLAATS :NUM :VAN :NAAR
MAKE "MAAT LIST (:NUM * :LENGTE) :HOOGTE
STEEN :NUM THING WORD "XST :NUM THING =>
 WORD "YST :NUM 0
MAKE WORD :VAN "TOP THING WORD "YST :NUM
IF EQUALP :NAAR "A [MAKE "XX -100]
IF EQUALP :NAAR "B [MAKE "XX 0]
IF EQUALP :NAAR "C [MAKE "XX 100]
STEEN :NUM (:XX - (:NUM * :LENGTE / 2 =>
)) THING WORD :NAAR "TOP 1
MAKE "BEURT :BEURT + 1
MAKE WORD :NAAR "TOP :HOOGTE + (THING =>
 WORD "YST :NUM)
END

TO STEEN :NUM :X :Y :K
SETPC :K
(LOCAL "VX "VY)
MAKE "GFILL "TRUE
SETFILL LPUT :K [2 2]
BOX SE :X SE :Y :MAAT
SETFILL [2 2 0]
MAKE "VX :X + ((FIRST :MAAT) / 2) - 8
MAKE "VY :Y + ((FIRST BF :MAAT) / 2) - 8
BOX SE :VX SE :VY [16 16]
PU SETPOS LIST :VX + 3 :VY + 2 PD
TT :NUM
MAKE WORD "XST :NUM :X
MAKE WORD "YST :NUM :Y
SETPC 1
END

TO BEGIN
FULLSCREEN CLEARSCREEN
MAKE "GFILL "FALSE
SETLINE [1 5 1]
BOX [-205 -50 410 100]
SETLINE [1 1 1]
BOX [-198 -45 396 90]
PU SETPOS [-150 10] PD
TT [* Welkom bij de Torens van Hanoi *]
PU SETPOS [-140 -25] PD
TT [Met hoeveel stenen zal ik spelen ?]
MAKE "AANTAL RC
END

TO EINDE
FULLSCREEN CLEARSCREEN
MAKE "GFILL "FALSE
SETLINE [1 5 1]
BOX [-205 -50 410 100]
SETLINE [1 1 1]
BOX [-198 -45 396 90]
PU SETPOS [-50 10] PD
TT [Opgelost !!!!!]
PU SETPOS [-110 -15] PD
TT (SE [Ik deed het in :] :BEURT "beu =>
rten "!)
PU SETPOS [-80 -35] PD
TT [Nog een keertje J / N]
MAKE "KEUS RC
IF EQUALP :KEUS "J [START]
END

TO ZET :NUM :VAN :NAAR :OVER
IF EQUALP :NUM 0 [STOP]
ZET :NUM - 1 :VAN :OVER :NAAR
VERPLAATS :NUM :VAN :NAAR
ZET :NUM - 1 :OVER :NAAR :VAN
END
```

Het zou al te gemakkelijk zijn deze vraag met een absoluut 'ja' of 'nee' te beantwoorden. Zeker is echter dat computers sinds lang geen "domme rekenapparaten" meer zijn. De bewering dat een computer "niets meer oplevert, dan er in is gestopt", doet dan ook zeker geen recht aan de huidige modellen. Maar kunnen computers dan denken? Om hier achter te komen zullen we eerst eens moeten nadenken over de manier waarop we zelf denken. Dat is haast recursief: denken over denken. Over het menselijke denken zijn boeken vol geschreven en dat willen we hier niet herhalen. Steeds komt het er echter op neer dat het menselijke denken "niet-formeel" en "associatief" wordt genoemd. Een voordeel van dit niet-formele en associatieve denken is dat mensen in hun denkproces heel kreatief kunnen zijn.

Ook wordt het menselijke denken wel "heuristisch" genoemd. Dit wil zeggen: niet systematisch, waarbij steeds nieuwe ingangen worden gezocht. Mensen denken dus vaak onlogisch, van de hak op de tak en gevoelsmatig. Daarbij laten we ons sterk leiden door eerder opgedane kennis (of emoties). Desondanks (of misschien "dank zij") zijn wij in staat gegevens te interpreteren. En dat is iets wat computers voorlopig nog niet kunnen.

Voor zover computers denken, is dit zuiver algoritmisch (welhaast wiskundig) van aard. Een computer is in staat kontinu (zonder invloeden van buitenaf) zeer grote hoeveelheden gegevens te bewerken. Op volkomen logische en systematische wijze is een computer in staat een oplossingsmethode te "bedenken".

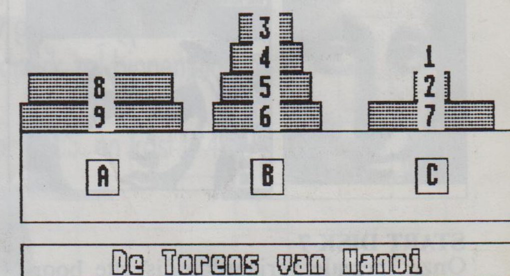
Het algoritmisch denken van computers wordt vaak geïllustreerd met een "boomstruktuur". Stelt u zich de stam van de boom als de ingang van het probleem voor (de probleemstelling) en de (soms miljoenen) twijgen als "mogelijke oplossingen". Door alle mogelijkheden en onmogelijkheden af te lopen, bereikt de computer uiteindelijk zijn doel. Een aardige illustratie van het logisch en analytisch denkvermogen van computers zien we onder andere bij schaakprogramma's. Volgens mensen die het kunnen weten zijn er op een schaakbord $10^{120}$ zetten mogelijk. Door het vroegtijdig elimineren van onlogische en ongewenste zetten, hoeft het programma echter niet al deze mogelijkheden na te lopen en kan het zijn "bedenktijd" aardig inkorten.

## STERREN

Iets dergelijks zien wij in een puzzel die "De Torens van Hanoi" wordt genoemd. Deze eeuwenoude oosterse puzzel bestaat uit een plankje met daarop drie stokken (A, B en C). Om de eerste stok zitten 64 schijven die steeds kleiner worden. De grootste schijf ligt onder en de kleinste ligt boven. De bedoeling van de puzzel is dat de schijventoren van A naar C wordt overgebracht, waarbij B als intermediair mag worden gebruikt. Er mag maar één schijf tegelijk verplaatst worden en een grote schijf mag nooit op een kleinere liggen. Als de puzzel is opgelost zullen (volgens het verhaal) de sterren van de hemel vallen. Ter geruststelling: voor de oplossing van deze puzzel zijn $2^{64}$-1 juiste zetten nodig. Een iets eenvoudiger uitvoering van de puzzel gebruikt een kleiner aantal schijven. Bij tien stuks zijn er echter nog altijd $2^{10}$-1 (= 1023) juiste zetten nodig.

In de listing ziet u staan hoe uw ST computer dit probleem op volledig logische wijze oplost. In een bekend (o.a. door Brian Harvey beschreven) algoritme dat is verwerkt in een meervoudig recursieve procedure (ZET) produceert het programma de oplossing van het probleem. Door het steeds verwisselen van de (lokale) variabelen worden onmogelijke zetten iedere keer uitgeschakeld. Bijvoorbeeld: als schijf 1 van A naar B wordt verplaatst, kan schijf 2 (die onder 1 lag) alleen naar C worden verplaatst. Door dit algoritme uit te werken kan de procedure voor u de oplossing van Hanoi "bedenken". Door middel van de procedures START, STEEN en VERPLAATS kunt u ook de grafische voorstelling van dit probleem op de voet volgen. Het gehele programma wordt gestart met de procedure BEGIN. Hierin vraagt het programma hoeveel schijven toren A moet bevatten. De procedure is geschreven voor gebruik met niet meer dan negen stenen. Een kleine aanpassing breidt het aantal stenen en daarmee het aantal mogelijkheden echter uit.

De kleinste hoeveelheid zetten die nodig is om het probleem op te lossen, is steeds gelijk aan $2^{N}$-1 (waarbij N het aantal stenen voorstelt). Ook anderszins is de listing te veranderen. U kunt bijvoorbeeld de routine ZET aanpassen zodat niet de computer, maar u de oplossing voor het probleem moet bedenken.




Standaard is echter voor de "denkende computer" gekozen om u te laten zien dat door logisch, systematisch en algoritmisch denken (en gebruik te maken van de krachtige mogelijkheden van Logo) uw ST in staat is een probleem op te lossen. Tevens illustreert deze listing nogmaals dat Logo ABSOLUUT geen kindertaal is, ook al spelen we in dit geval met torentjes. Kijk maar eens goed naar de fraaie struktuur van de oplossingsmethode.

*Auke Sikma*

# EN WAAR BLIJFT NU **ONS** KADOOTJE?

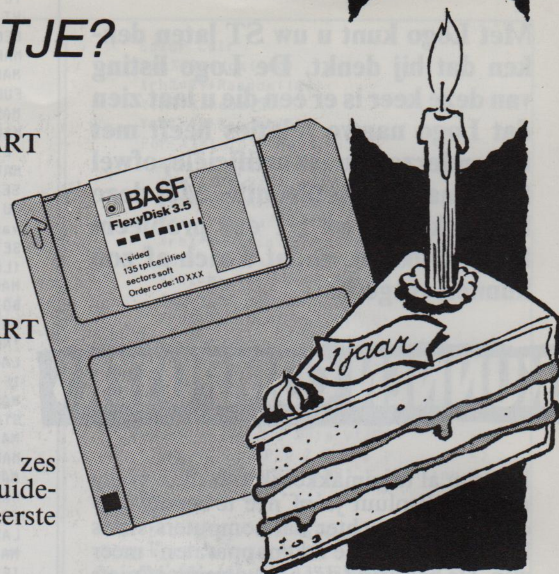
Op een verjaardag kun je niet met lege handen aankomen, dat spreekt. We maken het u niet moeilijk. Bovenaan ons verlanglijstje staat eigenlijk maar één ding: uw abonnement.
Zodra uw betaling binnen is, zenden we u onze grandioze verjaardagsdiskette. Zes klinkende START's heeft u daarna tegoed.

Een prima aanbod? Dachten we ook! Maar met een beperkte looptijd. De gratis diskette krijgt u alleen als uw overschrijving voor 8 november 1987 binnen is. Dus pak meteen dat giroblok en zorg dat we straks slingers van alle overschrijvingen kunnen maken!

Maak f 39,95 over naar
postgiro 1962230
t.a.v. Divo computerservice/START
M. Gijzenburg 14
2907 HG Capelle a/d IJssel

Voor Belgie:
Maak Bfr 800 over naar
postgiro 000-1592758-18
t.a.v. Divo computerservice/START
M. Gijzenburg 14
2907 HG Capelle a/d IJssel
Nederland

Een abonnement bestaat uit zes opeenvolgende bladen. Geef duidelijk aan welk nummer u als eerste wenst te ontvangen.




## START LEZERSSERVICE

De listings in elk nummer van START zijn ook op diskette verkrijgbaar. U bespaart uzelf veel tijd voor betrekkelijk weinig geld.
Listings in een compiler-afhankelijke programmeertaal worden ook in hun uiteindelijke vorm op de START-diskette gezet.
Als u zelf niet over een compiler beschikt en het programma toch graag in uw bezit wilt hebben, kan dit een uitkomst zijn.
Bovendien brengt START onder de naam SPECIAL DISKS ook een laaggeprijsde reeks software van Nederlandse auteurs uit.

### NIEUW:

**START SPECIAL DISK 6: TOOLKIT GEM**
(auteur: Egon Lubbers)
Bibliotheek met GfA-routines waarmee u spelenderwijs vensters, dialoogboxen, ikonen en dergelijke aan uw eigen programma's kunt toevoegen.




**START DISK 7**
Onze speciale verjaardagsdiskette boordevol verrassingen.

**Verder zijn nog verkrijgbaar:**

**START SPECIAL DISK 1: ELDICAD**
(auteur: Rob Hoo)
Programma waarmee onder muisbesturing elektronikaschema's gemaakt kunnen worden. Hoge resolutie vereist.

**START SPECIAL DISK 2: KATALOGER**
(auteur: Heino Kaal)
Programma waarmee u al uw directories kunt verzamelen, sorteren en afdrukken. Voor hoge of middelste resolutie.

**START SPECIAL DISK 3: PAINT**
(auteur: Anton Raves)
Ikoon gestuurd tekenprogramma voor alle resoluties. Compatibel met DEGAS, Art Director, Doodle en Neochrome.

**START SPECIAL DISK 4: SPELLBOUND**
(auteur Rob Koopman)
Spellingschecker met uitgebreid Nederlands woordenboek. Kontrole tijdens het typen of achteraf. Toevoegen aan woordenboek mogelijk. Voor hoge of middelste resolutie.

**START SPECIAL DISK 5: SAPRISTI**
(auteur A. Bollongino)
Etikettenmanipulator met mogelijkheden voor adreslabels, diskettelabels en eigen ontwerpen. Inklusief database en RAM-disk. Voor hoge of middelste resolutie.

**START DISKS 1 t/m 6**
Alle listings en programma's uit het blad met hetzelfde nummer.

**A. SCREENDUMP**
(auteur: Rob Koopman)
Uitgewerkt voorbeeld in Modula-2 om een eigen screendumproutine te schrijven.

**B. MEGAMAX**
(auteur: Egon Lubbers)
Programma's en sources van:
Impossible Figures, Mastermind, Life en Music Construction Set.

**C. MAKE**
Utility waarmee programmeren in C gemakkelijker wordt gemaakt. "Make" stuurt alleen gewijzigde programmadelen naar de compiler.

**D. ARC**
Utility waarmee u uw programma's in sterk verkorte vorm kunt bewaren. Ideaal voor een zo klein mogelijk aantal backup-schijven.

**HARDWARE: TELTRON EXTENDER**
Schakelkastje waarmee het populaire Teltron C64 modem op de ST aangesloten kan worden. Inklusief voorbeeldprogramma softwarematige baudrate instelling en autodial.

**PRIJZEN:**

| | |
|---|---|
| **DISKS 1 t/m 7, A t/m D:** | |
| Abonnees | f 15,--/Bfr 280 |
| Niet-abonnees | f 25,--/Bfr 460 |
| **KATALOGER, PAINT, SAPRISTI:** | |
| Abonnees | f 39,--/Bfr 720 |
| Niet-abonnees | f 49,--/Bfr 910 |
| **ELDICAD, SPELLBOUND, TOOLKIT:** | |
| Abonnees | f 49,--/Bfr 910 |
| Niet-abonnees | f 69,--/Bfr 1280 |
| TELTRON EXTENDER | f 199,--/Bfr 3680 |
| START (oude nummers) | f 6,95/Bfr 140 |

**BESTELWIJZE:**
Maak het verschuldigde bedrag over op postgiro 1962230
t.a.v. Divo Computerservice/START
M. Gijzenburg 14
2907 HG Capelle a/d IJssel
onder vermelding van de gewenste produkten.

Voor Belgie:
postgiro 000-1592758-18
Gebruik hetzelfde adres, maar voeg toe: Nederland.

## KATALOGER NU OOK IN KLEUR

Auteur Heino Kaal heeft onze tweede specialdisk nog eens stevig onder handen genomen. De volgende klinkende uitbreidingen zijn het resultaat:

- KATALOGER kan voortaan ook met een kleurenmonitor gebruikt worden (middelste resolutie)
- KATALOGER inventariseert een eventuele harde schijf
- KATALOGER leest desgewenst volautomatisch alleen hoofdprogramma's (PRG, TOS, TTP en APP bestanden) in

De prijs van KATALOGER blijft ongewijzigd:
abonnees f 39,-
niet-abonnees f 49,-
Geregistreerde gebruikers kunnen voor f 10,- een update bestellen. Vermeld hierbij s.v.p. uw serienummer.